Proposal for Design, Develop, Implement, and Maintain Comprehensive Support Portal for Singtel, Singapore

Created By:

**P. Ravindra Kumar (PMP)**

Senior IT Architecture Consultant

**Opensource Pte Ltd**

[1. Executive Summary 3](#_Toc456827772)

[2. Proposed Solution - Infrastructure 3](#_Toc456827773)

[2.1. Architecture Design Consideration 3](#_Toc456827774)

[2.1.1. Deployment Architecture 3](#_Toc456827775)

[2.1.2. Single Function for all Servers 4](#_Toc456827776)

[2.1.3. Secure Zone 4](#_Toc456827777)

[2.1.4. High Availability 4](#_Toc456827778)

[2.2. Infrastructure Implementation Approach 5](#_Toc456827779)

[2.2.1. Installation and Configuration 5](#_Toc456827780)

[2.2.2. UAT and Commission 5](#_Toc456827781)

[2.3. Infrastructure Overview 5](#_Toc456827782)

[2.3.1. Production Environment 5](#_Toc456827783)

[2.4. Backup Tool and Recommendation 6](#_Toc456827784)

[2.4.1. Backup Software – Scripted 6](#_Toc456827785)

[Features 6](#_Toc456827786)

[2.4.2. Recommendation – Proposed Backup Policy (Production) 7](#_Toc456827787)

[2.5. Scope of Work 8](#_Toc456827788)

[2.5.1. Proposed Solution 8](#_Toc456827789)

[System Objective 8](#_Toc456827790)

[Overview of Proposed System 8](#_Toc456827791)

[Solution Architecture 9](#_Toc456827792)

[Proposed Hardware 9](#_Toc456827793)

[Indicative Hosting Infrastructure 9](#_Toc456827794)

[2.6. Documentation 11](#_Toc456827795)

[2.7. Patch Management 11](#_Toc456827796)

[2.8. Failover scenario description 11](#_Toc456827797)

[2.9. PGP Support and Maintenance 12](#_Toc456827798)

[2.9.1. PGP Support 12](#_Toc456827799)

[2.9.2. Maintenance 12](#_Toc456827800)

[2.10. Implementation Methodology 12](#_Toc456827801)

[2.11. Assumptions 15](#_Toc456827802)

[2.22 Team Structure 16](#_Toc456827803)

[2.22.1 Team Roles and Responsibilities 17](#_Toc456827804)

[2.22.2 Team availability and further information 17](#_Toc456827805)

[3. Commercials: 18](#_Toc456827806)

[3.1Payment Terms: 19](#_Toc456827807)

[Dependencies and Constraints 19](#_Toc456827808)

[Authorization and Approval of Quotation 20](#_Toc456827809)

# Executive Summary

Opensource has been invited to participate in a RFQ involving Design, Develop, Implement, and Maintain a Web-based Comprehensive Support Portal for Singtel. Opensource is a leading IT service provider and systems integrator in Singapore providing a single point of contact for a wide range of professional and technical services customized to suit different organizational needs and budgets. With this contract, we hope that Customer will gain a better understanding of our business model and feel more confident working with us.

# Proposed Solution - Infrastructure

Opensource will be responsible for the design, implementation of Items as included in the infrastructure solution as follows:

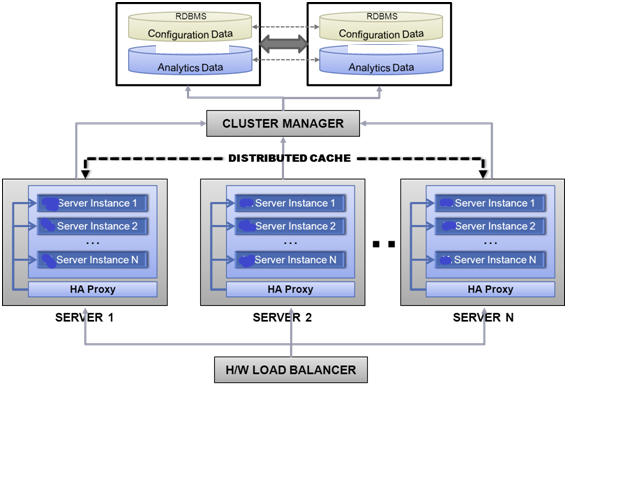
* Web server
* Application server
* Database server
* Hosting services for the aforementioned servers with high bandwidth connectivity to support multiple concurrent sessions
* High availability to ensure resiliency
* Firewall
* Load balancer
* Backup and disaster recovery for all the components
* Performance monitoring

# Architecture Design Consideration

Architecture design is done based on the below key factors.

# Deployment Architecture

Portal servers are linearly scalable. Portal Solution instances are mostly stateless, backed by centralized and replicable instances of RDBMS storage and (column storage) and fronted by High availability proxy server and hardware load balancers. The only state-full nature of the server is the authentication state, i.e. login sessions, which are by default distributed cache enabled. So, a new Portal server instance can be added to the existing cluster seamlessly, to scale up the performance.



### Single Function for all Servers

Proposed design is single function per server.

* Web Server will handle web requests
* Appl Server will handle application related functions including integration with ServiceNow
* Provide ACL based access to ensure confidentiality, data integrity and accountability
* Database will have redundant instances to prevent downtime

### Secure Zone

Proposed design is DMZ and Secure Zones

* Web servers will be placed in DMZ zone
* Rest of the servers (Application, database servers) will be placed in Secure Zone

### High Availability

Proposed design will handle failover and load balancing

* Two web servers to handle web requests
* Two app servers to handle application related functions and grouped under one cluster
* Two DB servers with bi-directional replication to handle DB requests

## Infrastructure Implementation Approach

## Installation and Configuration

* Install and Configure the software Firewall (apply the standards and policy)
* Install and Configure the software load balancer (apply the standards and policy)
* Install and Configure the Web Servers (apply the standards and policy)
* Install and Configure the App Servers (apply the standards and policy)
* Install and Configure the Database Servers (apply the standards and policy)
* Install and Configure the backup (apply the standards and policy)

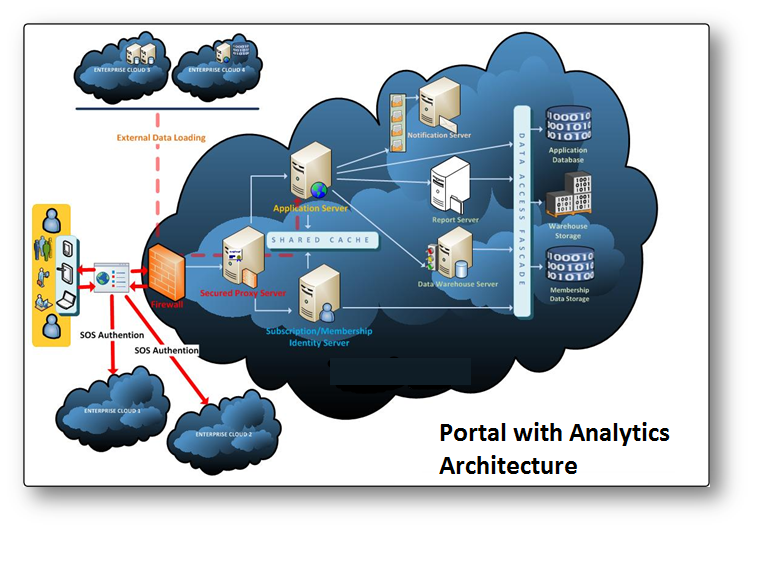
### UAT and Commission

* OS, Backup, Application
* Present Test results
* Hand-over

## Infrastructure Overview

### Production Environment

Production Environment Schematic Diagram:



The setup for the production environment is as follows:

For the production environment, the application server will be configured in 2 nodes of running instances of HTTP service for connectivity to ServiceNow.

The web server in turn will load balance over a cluster of 2 servers.

For the database, we will use MsSql Database in a 2 nodes setup with bi-directional replication.

The active/active architecture proposed supports fault-tolerance and high availability that is able to meet the system availability requirements stated in the tender.

RedHat Enterprise Linux 7 will be used as standard operating system. High level configuration details of the physical servers are given below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Production Servers | Tier | Qty | OS | CPU | Memory | Disks | NIC |
| Web Server | Web | 2 | RHL 7 | 2 vCore | 16 GB | 1 X 100 GB | 2 x 1 GB |
| Application Server | App | 2 | RHL 7 | 2 vCore | 16 GB | 1 X 100 GB | 2 x 1 GB |
| MsSql DB Server | DB | 2 | Win 2012 | 2 vCore | 16 GB | 1 X 100 GB | 2 x 1 GB |
| Firewall and Load Balancer Server |  | 2 | RHL 7 | 2 vCore | 8 GB | 1 X 50 GB | 2 x 1 GB |

## Backup Tool and Recommendation

### Backup Software – Scripted

A series of Shell and Perl scripts will be provided to perform daily scheduled backups of various components.

### Features

* Single solution for virtual and cloud environments.
* Simplifies backup and reduces cost and complexity associated with multiple point products.
* Protect more data while minimizing the disk space required for storing backup files.
* Backup to virtually any storage device including disk and cloud.

### Recommendation – Proposed Backup Policy (Production)

Complete file system backup will be taken for web, application and database servers. Online MsSql backup will also be conducted at regular intervals to ensure application consistent recoveries.

## Scope of Work

## Proposed Solution

### System Objective

E-Opensource will develop an application for CLIENT for automating its Employee Integration. The Solution will allow members of CLIENT anywhere anytime access to this Application with only a Computer and Internet/LAN connection without the need of installing any software.

When one plan to develop, acquire or revise a system they must be absolutely clear on the objectives of that system. The objectives must be stated in terms of the expected benefits that the business expects from investing in that system. The objectives define the expected return on investment.

A Self Service Portal Project has few primary business objectives:

* Design ,Develop and implementation of the optimization system
* Interfacing of all e-services with the proposed system
* Change Management
* Minimize service disruption to the smooth functioning of the daily operation.
* Minimize the disruption to the current ongoing implementation of the automation solution.

Minimize the impact on the proposed system, in the context of application changes, system configuration of the system, or testing effort required to ensure the proper functioning of the system.

* Ensure the delivery of high quality systems
* Provide strong management controls
* Maximize productivity
* In other words, the objective should ensure that we can produce more function, with higher

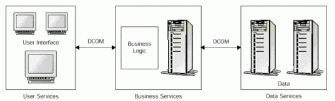
Quality in less time with proper resource and in a predictable manner.

### Overview of the Proposed System

The proposed web based system will be built on Microsoft .NET Technologies with N-Tier architecture. In system software, a web application is an application that is accessed over a network such as the Internet or an intranet. The term actually means a computer software application that is hosted in a browser-controlled environment or coded in a browser-supported language combined with a browser-rendered markup language like HTML) and reliant on a common web browser to render the application executable. Web applications are popular due to the ubiquity of web browsers, and the convenience of using a web browser as a client, sometimes called a thin client. Due to the limitations of the 2-tier client-server architecture, distributed applications are often divided up into three or more tiers. Components in each of these perform a specific type of processing – there's a User Services (Presentation) tier, a Business Services tier, and a Data Services tier in a 3-tier application.

The main advantage between this 3-tier architecture and traditional 2-tier client-server architecture is that, with 3-tier architecture, the business logic is separated from the user interface and the data source.

Breaking up applications into these separate tiers or sections will reduce the complexity of the overall application, and results in applications that can meet the growing needs of today's businesses. N-tier applications are just 3-tier applications that can further sub-divide the standard User Services, Business Services, or Data Services tiers.



In this type of application, the client should never access the data storage system directly. If it did, it would be circumventing the business rules of the application and would thus be unable to ensure that the data on display to the client was correct.

The separation of the various aspects of an application into n tiers allows for any part of that application to be modified without having to change the other parts, allowing developers to specialize in designing and developing a specific tier or tiers.

### Solution Architecture



**Cloud Brokerage Portal**

The portal presents an opportunity for clients to budget, select, configure, launch, review and maintain services. Importantly, in this aspect is the IAM which will be facilitated by Active Directory role based access. The idea is to give considerable freedom for developers to chose the application stack and infrastructure while ensuring the cost of the project is within the approved limits of the project. Thence the requirement defines the lifecycles of different entities:

* Project Life cycle (Propose, Estimate, Budget, Execute, Maintain, Migrate and Close)
* User Life Cycle and Associated Roles
* Virtual Data Centre Provider (IAM, Resource Provisioning, Security Groups descriptions)
* Service Catalogue (Service: Templates to define and jumpstart micro services components)

**Microsoft System Centre Orchestrator** This is used for WorkFlow Automation solution in the Data centre. While SCO will be utilized to run the automation core engine. ServiceNow Service Management will use Orchestration described briefly below:

* Setup workflow Roles based on IAM
* Use Orchestration to Automate complex task. E.g Request Password Reset Incident, or provisioning of scaling out services based on event like CPU 100% utilization.
* SAN provisioning using Incident Alerts workflow.

### Integration Using Integration Packs: The use of Integration packs is to integrate and extend the ServiceNow capability to CRUD Alerts/Incidents with Resource Service Providers viz Compute, Storage, Networking and Memory based on Templates services. Broadly speaking, ServiceNow Integration pack extends the notion of a service and uses the Infrastructure Integration Packs to provision, reclaim and update the resources ownerships in the ecosystems. These are defined below:

### ServiceNow Integration Pack: Use Kelvirion packs for ServiceNow to:

* Manage services to a CSP provider
* Execute runbooks based on Alerts

### Automate the following IT-Functions as per the roles and runbooks execution. The following will be defined and re-used for the following functions

* **Service Desk**

This typically will involve using standard and customized forms to create, read, update and delete incidents

* **Monitoring**This is triggered runbook execution due to Alerts generated at ServiceNow
* **Configuration Management**Import sets and templates to execute automate creation of scripted web service

### The infrastructure provider Integration packs have these goals: 1) To give ease of use for the common scripted workflows in an automated environment

2) To have plug-ins based implementation so that subsequent heterogeneous packs can installed and developed.

3) Generate Incidents based on mal-functioning of components.

4) Provision for logs to be used for Analytics to the Dashboard.

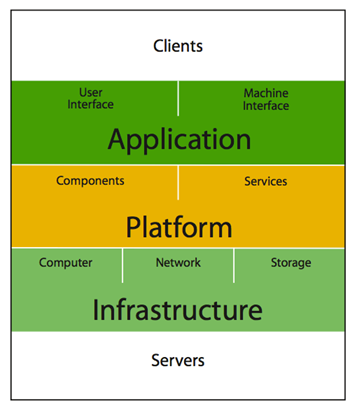
5) Monitoring Resource components as allocated by the Infrastructure vendors.

The following Integration Packs will be supported:

NetApp Integration Pack, EMC Integration Pack, Brocade Integration Pack, Vmware Integration Pack.

### Proposed Hardware

We propose Web Server and Application Server to be in the same Machine while RDBMS is preferably a separate server. The optimum configuration for the Web Server in Development Arena is given below in Hosting Infrastructure:



### Indicative Hosting Infrastructure

* Enterprise-Grade Rack mount Servers
* Powerful Dual Intel West mere-EP CPUs - 16 Cores
* Minimum of 12288MB (12GB) ECC DDR3 RAM
* Hot-Swappable(Like Seagate Cheetah 15,000 RPM) Serial Attached SCSI (SAS) Hard Disks
* Hardware RAID10 Disk Controllers.
* Use of 2U chassis for better cooling and air flow
* Redundant Power Supply Units (PSUs) for CDP and NAS backup servers
* Intelligent Platform Management Interface (IPMI) cards for hardware event monitoring
* (Remote reboot functionality allows team to power cycle servers faster if the need arises)

The following shall be provided as part of this project:

* Environment Setup
* Planning
* Confirm the required connections
* Confirm the required ports
* Confirm the required protocols
* Prepare the Technical Design Document
* Present the Technical Design document and sign-off
* Installation and Configuration
* Install and Configure Firewalls
* Install and Configure Load Balancer
* Install and Configure OS in all servers
* Configure all the network connections
* Configure NTP in all equipment
* Application Setup
* Planning
* Confirm the webserver configurations
* Confirm the application server configurations
* Confirm the high availability configurations
* Confirm the MsSql configurations
* Prepare the Technical Design Document
* Present the Technical Design document and sign-off
* Installation and Configuration
* Install and configure Web Servers
* Install and configure Analytic Servers
* Install and Configure Application Server
* Install and configure MsSql Servers
* Configure connectivity to the MsSql database
* Configure connectivity with ServiceNow
* Perform integration with User and Incident Management Module of application and ServiceNow accounts.
* Backup Server
* Confirm the Policy
* Installation and Configuration
* End-to-End Testing
* Failover Testing
* Training

## Documentation

* Plan Document
  + Technical Design Document
  + Infrastructure architecture
  + Virtualization design and specification
  + Network design and interface
* Installation and Configuration Document
* Test Pan
* Test Case Documents
* Trace-ability Matrix
* Test Results
* Test Summary Report
* Test Run Log

## Patch Management

Patch management is manual process and will be conducted on the passive machines first so that the impact to the BAU is minimal and pre-planned.

## Failover scenario description

|  |  |
| --- | --- |
| Failover Type | Description |
| Firewall (software) | Proposed solution consists of pairs of firewall. In case one firewall fail another firewall will take over the job. |
| Load Balancer (software) | Proposed solution consist of a pair of load balancer. In case one load balancer fail another load balancer will take over the job |
| Web Server | Proposed solution contains two web servers, if one web server fail, load balance will redirect all the requests to second/other web server |
| App Server | Proposed solution contains two App servers (application and tokenization), both servers are clustered. If one App server fail, web server requests will direct to second/other web server as connection is based on cluster name |
| DB server | Proposed solution contains two DB servers; both servers will sync using replication. If one DB server fail, App server will connect to other DB server |

## PGP Support and Maintenance

### PGP Support

Opensource will provide 3 months PGP for the implemented solution. Below activities will be performed during Warranty:

* Issue Support
* Fine tuning the system if there is any performance issue
* Patches (Patches will be applied quarterly)
* Software Upgrade Services (Upgrade will be performed once every 6 months)
* Documentation Correction (if required due to patches, upgrade or fine tune)
* Script Modifications (if required due to patches, upgrade or fine tune)
* Hand-over the system at end of the warranty

**Note: -** If any critical patches or updates that are required to be applied due to security reason or functional failure, ad-hoc work can be scheduled.

### Maintenance

Maintenance will be valid for a period of 3 years from the date of completion of 3 months PGP. Below activities will be performed during Maintenance:

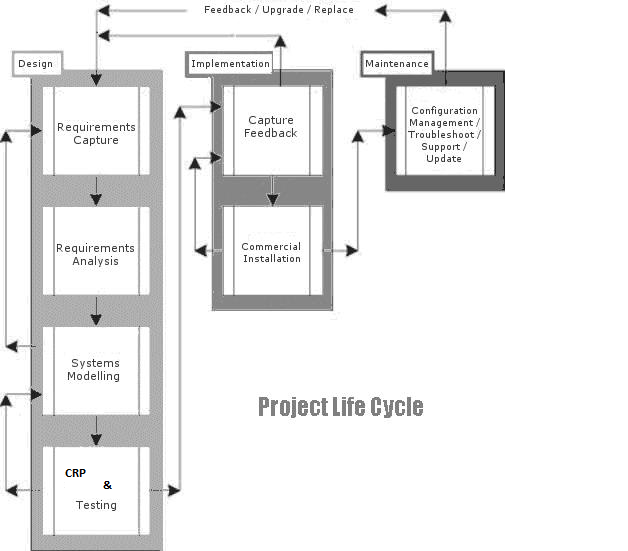
* Issue Support
* Preventive maintenance (will be performed every half yearly)
* Fine tuning the system for any performance issue
* Patches (Patches will be applied quarterly)
* Software Upgrade Services (Upgrade will be performed yearly once)
* Documentation Correction (if required due to patches, upgrade or fine tune)
* Script Modifications (if required due to patches, upgrade or fine tune)
* Recover lost data, restoration and repair of damaged data
* Maintain the inventory list

**Note: -** If any critical patches or updates that are required to be applied due to security reason or functional failure, ad-hoc work can be scheduled.

## Implementation Methodology

E-OpenSource proposes agile methodology for implementation. The CoE team in Singtel can identify the requirements, set priority based on “ready” or “to-be-ready” state of departments/functions and plan to implement those as “Unit Projects”.

Following schematic diagram shows the implementation lifecycle of a “Unit Project”.



There will be two separate systems, one for staging and another for production. All configurations will be implemented in the staging system first and will be tested by both, the teams from E-Opensource and Singtel. After successful completion of testing, the configuration will be replicated in the Production system.



We recommend the E²IGO (Evaluation > eLearning > Implementing > Go-Live > Optimization) implementation methodology for business applications

**Phase 1: Evaluation**

* Publish the latest and version of Portal release
* Analyze collected requirements to filter top-10 requirements.
* Define project scope (what to do and what not to do).

**Phase 2: eLearning (For Consultants)**

* Practice with examples guided by the intensive functional outcome .
* Practice with examples guided by the technical outcome

**Phase 3: Implementing**

* Define project schedule and resources, and identify project risks.
* Setup system and business rules, then initialize data and use of legacy data
* Customize and extend the solution to meet complex requirements.

**Phase 4: Go-Live**

* Train end-users concepts and usage with the help of Trainers
* After running legacy system and new system parallel for a few weeks, deploy the solution into production environment.

**Phase 5: Optimization**

* Collect special function requirements from end-users after the transition.
* Analyze the weakness of the solution and document the optimization requirements.

## Assumptions

* SINGTEL has to provide the data center access as necessary
* SINGTEL Design & Technical Document and other relevant templates should be provided if required
* SINGTEL shall be responsible for providing access to ServiceNow API’s (including REST API’s)
* Necessary superuser access will be provided
* All documents submitted to SINGTEL for sign-off will be considered as accepted by SINGTEL if there is no feedback one month after the submission date. SINGTEL will nominate an authorized person to sign off for each document
* Project will not go-live unless we have sign-off on UAT from all authorized SINGTEL personnel

## 2.22 Team Structure

We have assembled a team of experienced professionals for the completion of this engagement. Our proposed project structure of the core team members is as depicted in the figure below.

**Engineers**

1. Troy (OS, DB)
2. Ash (OS)
3. Ravi (Application, FW, LB)
4. Offshore Development Team

(coding, ServiceNow integration)

**Project Manager**

**Team Lead**

Ravi

Ankur

**Project Team**

### Team Roles and Responsibilities

1. Project Manager

* Responsible for overall delivery of project and ensuring that deliverables are up to

Opensource’s standards and customer’s requirements.

* Perform preliminary quality assurance reviews on all work on this project and endorse all formal deliverables.
* Primary contact for OPENSOURCE’s management.

1. Team Lead

* Prepare the project plan and approach, ensuring that these achieve the objectives of this tender.
* Monitor and report on the plan and make the necessary adjustments when circumstances arise.
* Manage the audit on a day-to-day basis.
* Manage engagement staffing requirements.
* Review the work performed.
* Review reports and documentation.

1. Engineer

* Execution of the testing and review work.
* Liaise with the various stakeholders and network implementation vendors as required in the course of the engagement.
* Produce reports and other supporting documentation.

### Team availability and further information

Opensource deserves the right to nominate new staff for OPENSOURCE’s consideration if any of the above nominated staff members become unavailable. As Opensource has a large team of staff, they would not have a problem nominating others who are equally suitable for the project.



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Commercials: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1Payment Terms: | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **#** | **Stage** | **Payment (% of total)** | **Cumulative Total %** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **1** | Upon acceptance of proposal | 50 | 50 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **2** | Mock up sign off | 25 | 75 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **3** | UAT and documentation | 25 | 100 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **4** | Annual maintenance | 100 upfront | n/a |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Dependencies and Constraints | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | The following assumptions were made while making the technical and commercial offer. | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1. Hardware and Infrastructure are to be available at the time of implementation. | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 2. Access to the third party hosted server will be available at the site implementation phase. | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 3. Architectural or core design change at the maintenance phase will not be taken care of. | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 4. Problem in existing data will be taken care separately. | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 5. Bulk of development in change request will be in offshore. | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 6. Onsite location is assumed to be the Client's premise in Singapore | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 7. Drivers, APIs etc for third party peripherals should be available at the time of integration. | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 8. Acceptance criteria for the project shall be as following : | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | ·         An Acceptance Test Plan will be prepared after SRS is signed-off jointly by NISHAN and Client  This will serve as the basis for Acceptance Testing Phase. | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |
|  | 9. The software customization and warranty support will cover any defects in the solution for which  E-Opensource is responsible. Any major or minor change of requirement, as frozen in the scope of work,  will not be treated as a defect and will carry a cost as per time and material basis.  The changes whether a defect or new requirement along with a cost implication,  will be governed by 'Change Control Process'. | | | | | | | | | | | | | | | | | | | | | | |  |
|  | 10. The Training described here in the schedule covers the Hands-on Training on the  developed application systems for the trainers. Here they will be made acquainted  with the application modules. | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# Authorization and Approval of Quotation

To: Opensource Pte Ltd Fax: 65-62344374

[  ] We hereby accept and approve the above quotation **(OSPL/Singtel/8984**)

Authorised by:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_ |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Signature |  | Company Stamp |  | Name |  | Date |

**--End of Document--**