**eService Information Package Design Document**

Author: Adam Wang

Contents

[1. Introduction 2](#_Toc425170639)

[1.1. Purpose 2](#_Toc425170640)

[1.2. Scope 2](#_Toc425170641)

[1.3. Overview 2](#_Toc425170642)

[1.4. Reference Material 2](#_Toc425170643)

[1.5. Definitions and Acronyms 2](#_Toc425170644)

[2. System Overview 2](#_Toc425170645)

[3. System Architecture 3](#_Toc425170646)

[3.1. Architecture Design 3](#_Toc425170647)

[3.2. Decomposition Description 4](#_Toc425170648)

[3.2.1. Domain Model 4](#_Toc425170649)

[3.2.2. User Interface Change 4](#_Toc425170650)

[3.3. Design Rationale 5](#_Toc425170651)

[4. Data Design 5](#_Toc425170652)

[4.1. Data Description 5](#_Toc425170653)

[4.2. Data Dictionary 6](#_Toc425170654)

[5. Component Design 6](#_Toc425170655)

[5.1. Local Cache Manager 6](#_Toc425170656)

[5.2. Job Assembler 7](#_Toc425170657)

[5.3. Tree View 7](#_Toc425170658)

[6. Human Interface Design 7](#_Toc425170659)

[6.1. Overview of User Interface 7](#_Toc425170660)

[6.1.1. Journal Management Console Tab 7](#_Toc425170661)

[6.1.2. Single Information Package View 8](#_Toc425170662)

[6.1.3. Information Package in Treeview 9](#_Toc425170663)

[6.2. Scree Objects and Actions 9](#_Toc425170664)

[7. Requirement Matrix 9](#_Toc425170665)

[8. Appendices 9](#_Toc425170666)

# Introduction

## Purpose

This software design document describes the architecture enhancement and system for eService. It will bring the awareness of system impact on other systems in Sanjel and demonstrate the potential capability of eService may achieve.

## Scope

The enhancement is only for eService artifacts structure addition. It will provide the foundation for eService delivering new functionalities such like separate information transmission, team collaboration, etc.

## Overview

This document provides domain model change, design model change and data change. Also describe the potential impact on other systems within Sanjel.

## Reference Material

## Definitions and Acronyms

# System Overview

eService was originally designed for Cementing service line, the job package is designed per job regardless how long the job lasts. Job Header, Service Ticket and Service Report are packed in one package and sent to HO at one time.

Currently eService cannot satisfy following business requirements:

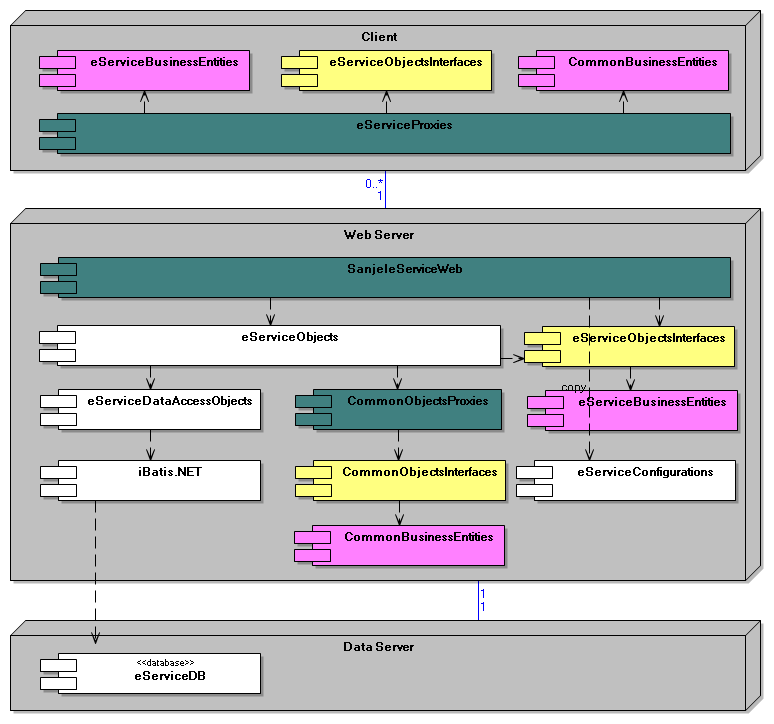
* Fracturing Service Line Job may last more than one day.
* Customer may request to be invoiced per day or per job.
* HSE requests operational quality data within 48 hours after incident occurs.
* HR needs to pay bonus based on daily revenue.
* Summary invoice needs to be approved by DSLM at the end.
* Daily invoice needs to be approved by DSLM per ticket.

The root cause is that job artifacts are tightly sealed in job package and being transmitted on time. To overcome the challenges, job artifacts need to be decoupled and associations will be maintained by constraints. Independent artifact transmission needs to be supported.

# System Architecture

## Architecture Design

eService was originally designed as “Transfer and store all third party Artifacts electronically as unstructured data in the corporate database.”, but during 8 years evolution, job artifacts had been implemented as tightly coupled package. However, the system architecture has not been changed.



## Decomposition Description

### Domain Model

To manage job artifacts properly, some domain entities need to be introduced in eService scope to enforce the constraints and maintain data integrity.



### User Interface Change

To achieve Information Package can be managed and transmitted separately; Journal needs to be viewed in two ways from local, by Journal or in Job Package.

* 1. Offline user interface
* Adding Journal From Local tab, view all information packages in a list.
  + Information package can be added/edited/removed from the list
  + Information package can be transmitted from the list
  + Transmitted information package can read-only
  + Information package can be sent from Journal From Local tab.
* Adding Information packages in Service Ticket and Service Report tree view
  + Information package items can be added/edited in the tree view
  + Transmitted information package can read-only
  1. Online user interface
* Adding Information packages in Service Ticket and Service Report tree view
  + Information package can be viewed/added/edited/removed on server in job package tree view.
  + Information package cannot be added/edited/removed after the job being approved.

## Design Rationale

While considering the architecture improvement, we are limited by following factors:

* No enough resource to support major architecture refactoring
* Major business process change cannot be considered
* Major user experience cannot be changed

So the stage approach is adopted, adding loose coupled artifact is first step to allow eService handle unstructured data. Further stages may allow us to move some artifacts out of Service Ticket and Service Report to achieve the flexibility of management and transmission.

# Data Design

## Data Description

Journal and information data are stored in same database with eService. The data modelling and access will follow existing eService framework patterns.

## Data Dictionary



# Component Design

Some components need to be enhanced to support the addition structure.

## Local Cache Manager

Currently major entities such like Call Sheet, Job, Service Ticket and Service Report are saved on local as one single file in a serialized xml format. There is no cross reference between each other.

To achieve loose couple objects aggregation, Local Cache Manager needs to be able to look up serialized objects to find cross references.

* 1. Message Executor

Message Executor works on server side to execute the operation defined in message package after it is received completely by Message receiver. Currently Message Executor can only identify 3 types of object, Job, Service Report and Service Ticket. New object type needs to be registered.

## Job Assembler

Job Assembler verifies received objects and update job package status. The job package structure change and related business rules need to be implemented.

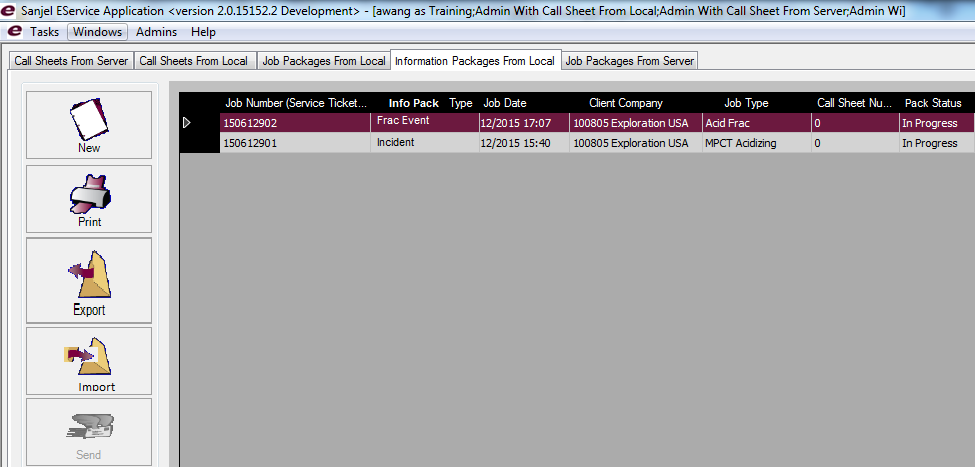
## Tree View

One major enhancement to eService is to add capability of multiple object views. Loose coupled object can be viewed independently and aggregately in another object. Currently Tree View component can only interpret single aggregated object which is aggregated on business entity level. One abstracted level view model needs to be introduced to aggregate objects from different source.

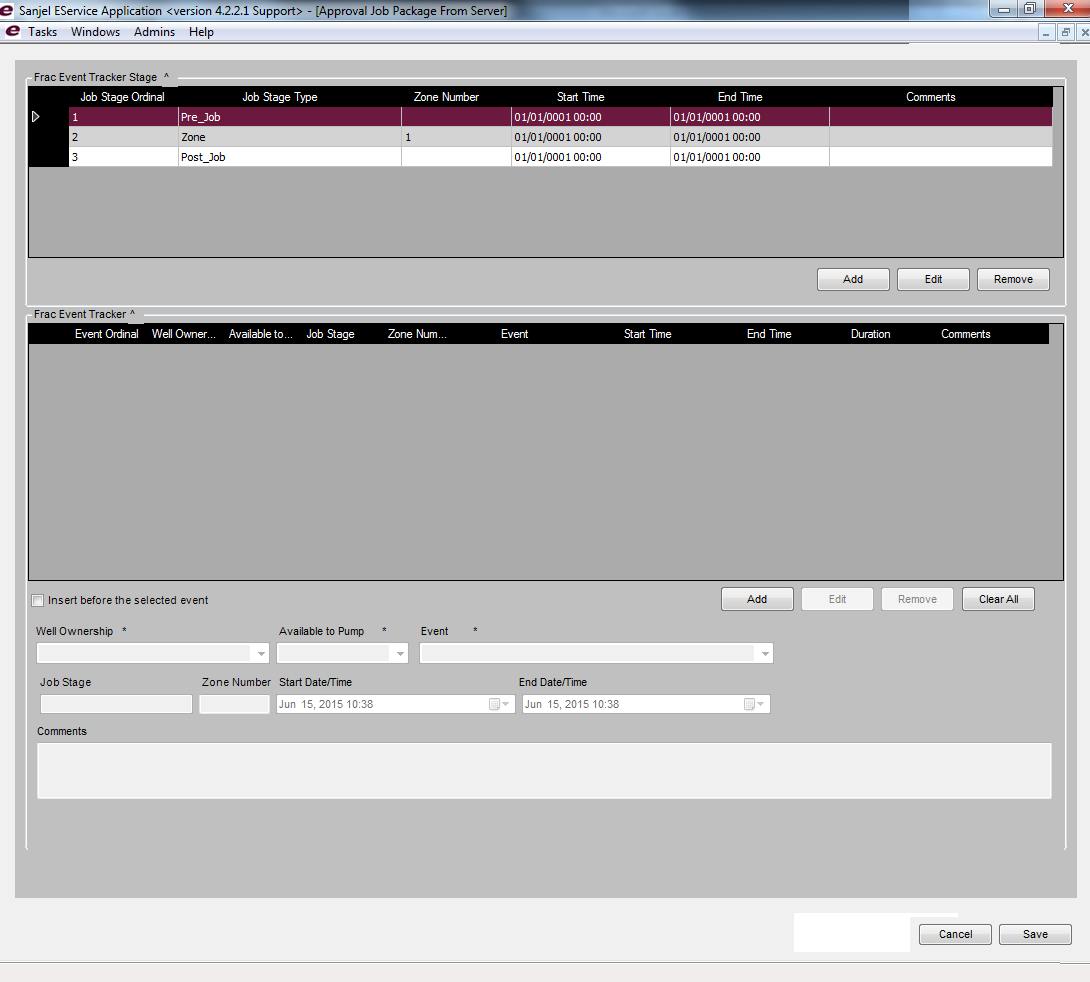
# Human Interface Design

## Overview of User Interface

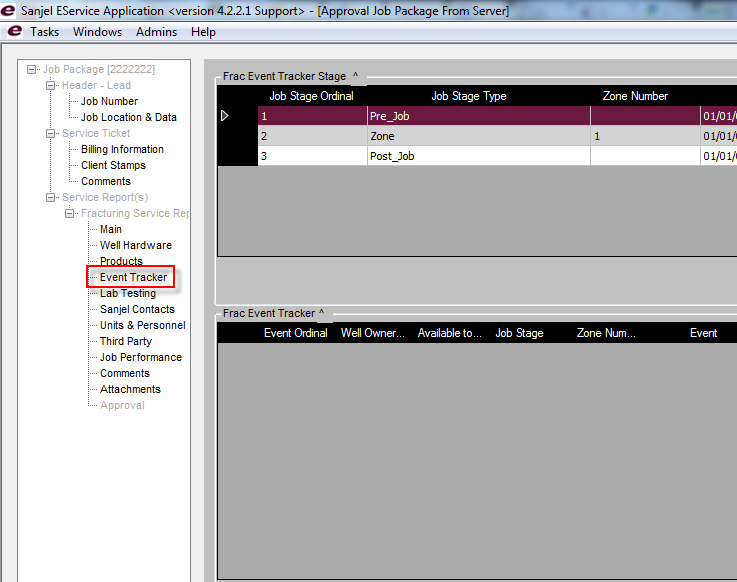
### Journal Management Console Tab



### Single Information Package View



### Information Package in Treeview



## Scree Objects and Actions

# Requirement Matrix

# Appendices