



**Unity Programme**

**UK – Release 4**

**Duplicate Cover Check Service**

**Technical Design**

**Version 0.3**

Table of CONTENTS

[1 Introduction 4](#_Toc440014518)

[1.1 MDM Platform Overview 4](#_Toc440014519)

[1.2 Objective 4](#_Toc440014520)

[1.3 Distribution List 5](#_Toc440014521)

[1.4 Glossary 5](#_Toc440014522)

[2 Technical Design 6](#_Toc440014523)

[2.1 Requirements Overview 6](#_Toc440014524)

[2.2 High Level Component and Dataflow 6](#_Toc440014525)

[2.3 Design Requirements 7](#_Toc440014526)

[2.3.1 Data Validation and Business Rules 7](#_Toc440014527)

[2.4 Assumptions 8](#_Toc440014528)

[2.5 Design Decisions 9](#_Toc440014529)

[2.6 Risks 9](#_Toc440014530)

[2.7 Issues 9](#_Toc440014531)

[2.8 Interface Details 9](#_Toc440014532)

[2.8.1 Interface Specification Details 9](#_Toc440014533)

[2.8.2 Interface Frequency and Schedule 10](#_Toc440014534)

[2.9 Flowchart 11](#_Toc440014535)

[2.10 Mapping and Transformation Rules 13](#_Toc440014536)

[2.11 Technical Component Details 14](#_Toc440014537)

[2.12 Service Availability Service 16](#_Toc440014538)

[2.13 Data Exception Reports 16](#_Toc440014539)

[3 Non-Functional Design 17](#_Toc440014540)

[3.1 Security and Data Access 17](#_Toc440014541)

[3.2 Performance Consideration 17](#_Toc440014542)

[3.3 Deployment 17](#_Toc440014543)

[3.4 Error Handling and Fault Management 18](#_Toc440014544)

[3.4.1 Fault Codes and Description: 18](#_Toc440014545)

[3.5 Housekeeping Policies 18](#_Toc440014546)

[4 Appendix 19](#_Toc440014547)

[4.1 DuplicateCoverCheck services ICD 19](#_Toc440014548)

**Document Maintenance**

***Document Owners***

|  |  |
| --- | --- |
| Functional owner(s) / Architect: Karthikeyan Ramanathan | |
| Technical Owner(s): Sekhar | |
| Current Version: | 0.3 |
| Date Last Updated: | 04/01/2016 |

***Version Control***

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date Updated | Revision Author | Summary of Major Changes Made |
| 0.1 | 28/05/2015 | Chethana | Initial draft |
| 0.2 | 10/06/2015 | Chethana | Updated section 2 as per the comments |
| 0.3 | 04/01/2015 | Sekhar | Updated error codes  Added design decision D05 |

**Note:** These documents are strictly for specific Virtusa/HomeServe use only. They shall not be shared with an external party other than HomeServe. These documents should always be kept securely and employees shall use reasonable care protecting these documents from unauthorized use or disclosure to a third party. This category also covers client intellectual property where Virtusa has a non-disclosure agreement with HomeServe.

**Document Approval**

Virtusa Corporation and HomeServe have reviewed this document and hereby agree that the contents herein are accurate. Any changes to this document must be communicated in writing and signed-off by both parties.

|  |  |
| --- | --- |
| Signature | Signature |
| Date: | Date: |
| Name: | Name: |
| Client: HomeServe | Virtusa Corporation |
| Address:  Cable Drive,  Walsall,  West Midlands  WS2 7BN | Address:  Virtusa (UK) Ltd  Level 8  26 Finsbury Square London EC2A 1D  E-mail: [info@virtusa.com](mailto:Info@techprov.com)  URL: [www.virtusa.com](http://www.techprov.com) |

# Introduction

As part of Release 4 there is a need to integrate active policy data in Ensura to MDM in near real time along with the contracts from PEGA to facilitate duplicate cover checks based on compliance requirements

## MDM Platform Overview

Master Data Management (MDM) comprises the processes, governance, policies, standards and tools that consistently define and manage the critical data of an organization to provide a single point of reference.

In Homeserve, the strategic platform uses MDM system to consolidate the customer/prospect information from various source systems for operational and marketing use. The following diagram outlines the overall MDM Platform architecture.



Calling systems PEGA/Ensura are integrated with MDM in real-time as part of Unity Release 4. The calling systems will use the same Talend ESB service to send the request to MDM for Duplicate Cover Check operation.

## Objective

Objective of this design document is to outline the capability of the service to identify the overlap/duplicates cover against the insured property. This document outlines the various technical aspects on how the request is consumed and also about generating the response through Talend ESB.

This document mainly aimed to support the construction phase and it covers the following details,

1. High level data integration flow
2. Data Validations and Business rules
3. Design Decisions and Assumptions
4. Interface details
5. Flowchart
6. Mapping specifications
7. Non-functional designs

## Distribution List

Approval List

|  |  |  |
| --- | --- | --- |
| **Name** | **Organization** | **Role** |
| Mark Buck | HomeServe | Solution Architect |
|  |  |  |
|  |  |  |
|  |  |  |

Discussion / Review List

In addition to those on the approval list above comments will be sought from the following people/groups.

|  |  |  |
| --- | --- | --- |
| **Name** | **Organization** | **Role** |
| Naveen Mula | Virtusa | Lead Data Architect |
| Alok Jha | Virtusa | Architect |

Governance

|  |  |  |
| --- | --- | --- |
| **Name** | **Applicable?** | **Approval Details** |
| Technical Design Authority | Y |  |
| Business Design Authority | N | NA |
|  |  |  |
|  |  |  |

## Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| MDM | Master Data Management |
| SOAP | Simple Object Access Protocol |
| XML | eXtended Markup Language |
| WSDL | Web Services Description Language |
| SAM | Service Activity Monitoring |
| TAC | Talend Administration Console |
| ESB | Enterprise Service Bus |
| DQ | Data Quality |

# Technical Design

## Requirements Overview

As part of Unity Programme, Release 4 will provide a capability to identify the overlap/duplicates cover against the existing contract if any for insured property using Duplicate Cover Check service.

From Release 4, eServe will be live with:

* + P&D
  + P&D Plus
  + HSC7
  + HSC8

The high level requirement is the need to ensure that any HomeServe Customer with an active policy does not have the same type of cover in within or across PEGA and Ensura.

## High Level Component and Dataflow

The following diagram explains the high level component and dataflow of source system and Pega/Ensura integration



Frequency of interaction – Real-time.

Protocol – Soap over Https

## Design Requirements

## Data Validation and Business Rules

The following business rules will be applied coming from source system.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref. | Data Element | Business Rule Overview | ErrorCode |
| MDM-BR01 | Wsdl validation | All data exchanged between Pega/Ensura and MDM should be aligned with ICD and format/mandatory fields should completely be based on XSD definition | Default Talend Out-Of-Box xsd validation error description will be thrown. |
| MDM-BR02 | Postcode | Cleansing Rules - Postcode should be converted to UPPER case and leading and trailing blank spaces to be trimmed.  The cleansed version of Postcode should be used while searching the records in MDM | NA |

## Assumptions

| Ref. | Design Decision |
| --- | --- |
| A01 | Request from the calling systems should align with the structure defined in the XSD. |
| A02 | Calling systems will do the Duplicate Cover check internally before using the MDM Duplicate cover check service |
| A03 | Duplicate cover check will be performed against the records that are present in the MDM at the point in time. |
| A04 | MDM do not support multiple partner linked to a single contract in current release. In future we will have some enhancement when partner behavior needs to be included |
| A05 | Offerid (PEGA), Product Code (ENSURA) will be passed in the Request for Offerid TAG |
| A06 | Pega should send Capscan cleansed address information in the request. |
| A07 | For requests from Pega, Address matching will be done using Postcode and DPS, else with CapscanAddressKey if available from Pega, else perform exact match as per the next point. |
| A08 | For requests from Ensura, the address details are first cleansed with Capscan. Address matching will be done using Postcode and DPS, else with CapscanAddressKey if available from Capscan, else perform exact match as per the next point. |
| A09 | Address exact match for Ensura/Pega requests will be based on at least Flat Number and/or Building Number and Postcode along with other available request attributes. |
| A10 | The 'Effective Start Date' value in the request will be checked against the 'Contract End Date' in MDM. For example, if a contract exist and its end date is greater than effective start date then it will be considered as a duplicate |
| A11 | If 'Contract End Date' is missing for an active policy in MDM then it will be considered for duplicate cover check. |
| A12 | For duplicate cover check, active contract status (In-Force, Suspended) and latest major version will be considered. |
| A13 | Calling system will consume the response and process the information to interpret the results published. |
| A14 | Assets and Partners data is not in the scope for Duplicate cover check criteria |
| A15 | Pega will use the UpdateParty service to update any changes to the contract (Dates, status) for all previous versions. |
| A16 | Product Spine BIX extract from PEGA will have the Claim Center Product Code and Ensura Product Code will be passed at Package Level |

**Note: Please refer to ICD-INT-25 for more assumptions Section 2.1**

## Design Decisions

| Ref. | Design Decision |
| --- | --- |
| D01 | Metadata will be stored in different schema in MDM and this can be accessed to perform the duplicate cover check |
| D02 | Duplicate cover check for Address move scenario will be handled using the Contract start date |
| D03 | Duplicate cover check to be designed to find the overlap at Ensura Product Level/PEGA Package level |
| D04 | Duplicate Cover Check service is to be configured at run time per requesting source system id to search source system data other than calling system or both systems (PEGA & Ensura) |
| D05 | Create Party Service for Pega is a real time service where MDM doesn’t cleanse the address sent by Pega as, Pega is expected to cleanse addresses before sending the details. In an even when Capscan is down, Pega might send an invalid address which will not be cleansed by MDM.  As a result Duplicate cover check will be performed on invalid address. The onus is on Pega to send proper address details for duplicate cover check. |

## Risks

|  |  |  |  |
| --- | --- | --- | --- |
| Risk No. | Description | Comments | Owner |
| 1 | MDM Service for Duplicate Cover Check not available | This might lead to same policy cover being created. This needs to handle by Homeserve data governance team using the reports that will be generated from MDM on daily or weekly basis. | Homeserve Data Governance team |

## Issues

| Sno. | Description | Resolution | Owner | Status |
| --- | --- | --- | --- | --- |
| 1 | Product Mapping Structure is not yet defined | Homeserve team should provide the product mapping | Homeserve | Open |

## Interface Details

## Interface Specification Details

Below is the link to the ICD for this service

[https://serveusa.sharepoint.com/sites/HomeServe/GlobalResources/GlobalProjects/Ensura/Programme%20Documentation/**INT25\_MDM\_DuplicateCoverCheck\_ICD.docx**?d=w2f2d1d399b3149b0aab74d255dc5d9e9](https://serveusa.sharepoint.com/sites/HomeServe/GlobalResources/GlobalProjects/Ensura/Archive%20%20documents/Release%204/A&D/INT25_MDM_DuplicateCoverCheck_ICD.DOCX?d=w2f2d1d399b3149b0aab74d255dc5d9e9)

[https://serveusa.sharepoint.com/sites/HomeServe/GlobalResources/GlobalProjects/Ensura/Programme%20Documentation/**INT36%2637\_MDM\_ProductMapping\_ICD.docx**?d=w32af0c580b7449a8b811180fb0b9f728](https://serveusa.sharepoint.com/sites/HomeServe/GlobalResources/GlobalProjects/Ensura/Archive%20%20documents/Release%204/A&D/INT36&37_MDM_ProductMapping_ICD.docx?d=w32af0c580b7449a8b811180fb0b9f728)

## Interface Frequency and Schedule

| Interface Name | Frequency |
| --- | --- |
| DuplicateCoverCheck | Immediate (real-time)  On demand |

## Flowchart



Below are details about the each process indicated in the above flowchart:

**Address Match**

Pega should send Capscan cleansed address information in the request. For requests from Pega, Address matching will be done using Postcode and DPS, else with CapscanAddressKey. if available from Pega, else perform exact match as per the next point. For requests from Ensura, the address details are first cleansed with Capscan. Address matching will be done using Postcode and DPS, else with CapscanAddressKey if available from Capscan, else perform exact match as per the next point. Address exact match for Ensura/Pega requests will be based on at least Flat Number and/or Building Number and Postcode along with other available request attributes.

**Active Contracts**

All the Contracts along with the SSID in MDM ContractRefrence table from both or other calling system (based on the configuration at run time) will be fetched from MDM

**Contracts for Duplicate Cover Check**

Contract that are found in the previous step will be limited to the contracts that are effective as on Expected Start Date being sent in the request. Only this contract will be considered for matching and finding the overlap.

**Linked Offers**

Find the offers linked to a contract. Basing on the SSID the value in the OfferId could be a PEGA Offer or the Ensura Product code.

**Linked Customer**

Find the related party basing on the Address and the Contracts found. Party information is required to provide this in the duplicate cover response

**Lookup Product Mapping**

Product mapping table is the reference data in MDM that will hold information about the product metadata from PEGA/Ensura/Claim Center.

* Active contracts (OfferID/EnsuraProductCode) found in the previous step will be joined to the product mapping.
* OfferID sent from the request will also join to the product mapping to find the Ensura product to perform the overlap

**Duplicate Cover Flag**

This flag is available at different levels i.e. Offer and Property level. Duplicate cover at offer level is derived if there is any overlap between the Ensura product in the contract to the looked up Ensura product from the product mapping. Property level duplicate cover flag is derived if atleast on the offer is flagged as duplicate

**Duplicate Address Flag**

This is to indicate if there are any active contract regardless of the effectiveness around the Expected start date on the system other than the calling system. If the calling system is PEGA and there are active contract in MDM for Ensura then this flag will be set to True else False and vice versa.

**Type**

This attribute in response is the indicator to say it this offer/Contract is a Customer holding (Contract) or the Offer (Quote) from the request

## Mapping and Transformation Rules

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Item | Mapping Document | Version |
| 1 | DuplicateCoverCheck |  | 0.2 |

**End to end data mapping logic for Duplicate Cover mapping logic**



## Technical Component Details

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Functional Component | Description | Key Development Guidelines |
| 1 | Talend ESB – Https Authentication | This component will be used to authenticate the incoming message using https – username and password. | Use Talend ESB out-of-box security option to enable the https authentication. |
| 2 | Talend ESB –  Connection Pooling | This component is used to open a MDM server connection. | tMDMConnection: opens a MDM server connection for convenient reuse in the current transaction  tOracleConnection: used to connect the Oracle Database |
| 3 | Talend ESB -  WSDL/XSD Definitions | The Xml schemas and WSDL definitions will be maintained within server for each service exposed from MDM using Talend ESB | Use Talend ESB perspective to create and maintain the XSD and WSDLs**.**  Use Talend ESB Runtime options to validate WSDL |
| 4 | Talend DI - Real-time Request Listener | Listens the Source Request SOAP from Source system and send the request to downstream | tESBProviderRequest: used to capture SOAP request from Source system and parse the request related attributes send it to downstream |
| 5 | Talend DI - Data Validation | Validate the data from the Source System Request which is XML | tMap: Validates the request attribute based on the validation rules define. |
| 6 | Talend DI - Data Mapping/Transformations | In this component we apply Transformation Logic to load and validate the data | tFixedFlowInput: generates as many lines and columns as you want using the context variables.  tConvertType: allows specific conversions at runtime from one Talend java type to another  tXMLMap: transforms and routes data from single or multiple sources to single or multiple destinations.  tHashOutput: This component loads data to the cache memory to offer high-speed access, facilitating transactions involving a large amount of data.  tMap: transforms and routes data from single or multiple sources to single or multiple destinations.  tJava: makes it possible to extend the functionalities of a **Talend** Job through using Java commands.  tSetGlobalVar:  allows you to define and set global variables in GUI  tFlowToItereate:  iterates on the input data and generates global variables  tOracleRow: used to execute the dynamic SQL generated using the columns from request and MDM Entities |
| 7 | Talend DI - Business Rules | This Component checks the business Rules by applying various validation checks (regular expression) | Rules should be applied on the attributes to validate the request attributes.  Eg: Validate/check the postcode attribute using matches (<pattern>) |
| 9 | Talend DI / MDM - Response Generator | Generates the Response on success/failure and send the messages to Ensura | tESBProviderResponse: Should be used to generate response in xml format with error code and error description.  tESBProviderFault: Should be used for generating error messages with code and description |
| 10 | Talend MDM Database | Load the data into the Target MDM Database | tOracleInput: reads a database and extracts fields based on a query.  tMDMInput: reads a database and extracts fields based on the conditions. |
| 11 | Talend DI - Functional Error handling | Functional errors are generally included in the every response to intimate the users about the functional error and send with proper error code and error message. | tESBProviderResponse: should be used to generate response in xml format with error code and error description. |
| 12 | Talend DI - Technical Error Handling | Errors are generally occurs when the component failed to execute like component failed inside a job | tESBProviderFault: used for generating error messages with code and description |

## Service Availability Service

The Service Availability operation needs to provide responses to availability requests, working either at a service level only (i.e. checking for an external endpoint) or a deeper level incorporating more of the lower tiers in the back end system where it can as mentioned below

* + Deep Ping = false, respond with a “0” resultcode response confirming the Talend ESB service is up and running
  + Deep Ping = true, MDM level DB Check to ensure MDM is up and running using a simple select sql and successful result of the SQL is a “0” resultcode response, a negative response needs details on the issue returned from MDM db, equivalent to outcome/resultcode/descriptions defined for the main operation.

## Data Exception Reports

The data exception reports can be subscribed (On-Demand/Daily) from MDM system to support the manual data governance functions.

Talend MDM Web User Interface (out-of-box feature), Talend Staging and MDM database access should be granted to Data Stewards to support the manual data governance functions.

Below are the reports that will be built on MDM

* Customer records matched across stacks (Pega and Ensura) report will be captured and available within MDM system.
* To identify Duplicate cover sold to a party at property level as report will be made available within MDM system. This report should be published on daily basis to support the operational activities

# Non-Functional Design

## Security and Data Access

The Real-time service will be hosted within HomeServe network. The protocol will be SOAP Over HTTPS and requests and response end points will be secured with **Username & Password** as per the HomeServe security policies. All required SSL certificates should be configured in all environments (TAC) to support HTTPS

The following steps should be taken to enable the http username and password authentication for each service,

1. SOAP request from source system header will have username and password as per the security policy. The header information expected from the calling system is detailed in the HomeServe confluence below

Refer to “SOAP Header Standards” in wiki here: <http://confluence.hgb.hs.int:8090/display/DEV/SOAP+Messaging+Development+Practices>

1. Source System will be responsible for the storing of the username/password pair required to access the service, this information should be stored securely and safely
2. Requests will be secured with **Username & Password** as per the Homeserve InfoSec based security policies, namely WS-Security UsernameToken usage in the SOAP headers.

## Performance Consideration

The MDM services will align with following performance requirements, the following key performance guidelines need to be considered while implementing the service to meet the above performance requirements. Also refer the Talend Development Guidelines and Best practices for more details.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Service | Number of Request per Hour | Expected Response Time per request |
| 1 | DuplicateCoverCheck | 1000 | 5 Sec |

## Deployment

All MDM services will be deployed and monitored using Talend ESB module. The services artefacts will be available in Nexus repository and using Talend Administrator Centre (TAC), these artifacts will be deployed in Talend ESB runtime. Notification will be configured within TAC to send an alert to service delivery team whenever services are not available.

## Error Handling and Fault Management

To assist client systems with exception handling, all functional/business logic related errors will be part of SOAP service response with an appropriate Error code and Error Description will be sent as per the Error Catalogue.

## Fault Codes and Description:

|  |  |  |
| --- | --- | --- |
| **ID** | **Technical Description** | **Support Description** |
| 105 | Technical Server/Job/Service error | Technical Server/Job/Service error 1)MDM Database connectivity issue 2)Standard Database Error 3)Internal Job Error |
| 106 | Invalid Source System Identifier | Invalid Source System Identifier |
| 107 | Mandatory fields are missing | Mandatory Fields missing from Request |
| 131 | More than one address found | More than one address found for the requested address |
| 133 | Quote not present in the product mapping reference | Quote not present in in product mapping tables |
| 135 | Address not found in MDM | Request address not found in MDM |

## Housekeeping Policies

Talend services actual request and response messages will be logged in Talend Administrator Console (ESB Job Conductor) using Service Activity Monitoring (SAM) feature. The housekeeping policy will be aligned with Talend ESB policy defined for HomeServe.

# Appendix

## DuplicateCoverCheck services ICD

[https://serveusa.sharepoint.com/sites/HomeServe/GlobalResources/GlobalProjects/Ensura/Programme%20Documentation/INT25\_MDM\_DuplicateCoverCheck\_ICD.docx?d=w2f2d1d399b3149b0aab74d255dc5d9e9](https://serveusa.sharepoint.com/sites/HomeServe/GlobalResources/GlobalProjects/Ensura/Archive%20%20documents/Release%204/A&D/INT25_MDM_DuplicateCoverCheck_ICD.DOCX?d=w2f2d1d399b3149b0aab74d255dc5d9e9)