**Maxim POS Polling Enterprise Service Bus Implmentation Service**

**EL-FY16-902**

**Sale/Master/Pricing Data Exchange**

Requirement Definition Document

Prepared by: Buzz IT Company Limited

Author: Steven Chen

Date: 02/12/2016

Version: 1.0

# Document Control

## Document History

| Version | Date | Author | Revision Remark |
| --- | --- | --- | --- |
| 1.0 | 02/12/2016 | Steven Chen | 1st draft |
| 1.01 | 13/12/2016 | Steven Chen |  |

## Document/Design Owner

| Name | Title |
| --- | --- |
| Steven Chen | System Analyst |

## Key Comments

| Name/Title |  | |
| --- | --- | --- |
| # | Comments |
| **Comment** |  |  |
| **Response** |  |  |

# Table of Content

[Document Control 2](#_Toc468468905)

[Document History 2](#_Toc468468906)

[Document/Design Owner 2](#_Toc468468907)

[Key Comments 2](#_Toc468468908)

[Table of Content 3](#_Toc468468909)

[1 Background 5](#_Toc468468910)

[1.1 Document Purpose 5](#_Toc468468911)

[1.2 Document Scope 5](#_Toc468468912)

[1.3 Document Audience 5](#_Toc468468913)

[1.4 Terms & Abbreviation 6](#_Toc468468914)

[1.5 Reference Materials 6](#_Toc468468915)

[2 Executive Summary 7](#_Toc468468916)

[3 Business Context & Concept 8](#_Toc468468917)

[3.1 Business Rationale, Objectives & Considerations 8](#_Toc468468918)

[3.1.1 Objectives 8](#_Toc468468919)

[3.1.2 Expected Business Benefits 8](#_Toc468468920)

[3.1.3 Considerations 8](#_Toc468468921)

[3.1.3.1 Assumptions 8](#_Toc468468922)

[3.1.3.2 Constraints 10](#_Toc468468924)

[3.1.3.3 Dependencies 11](#_Toc468468926)

[3.1.3.4 Issues 11](#_Toc468468927)

[3.1.3.5 Risks 11](#_Toc468468929)

[3.2 Value Propositions 11](#_Toc468468930)

[4 Business Process 12](#_Toc468468965)

[4.1 Business Process Scope 12](#_Toc468468966)

[4.2 Sales Data 14](#_Toc468468967)

[4.2.1 Sales Data Overall Process Flow 14](#_Toc468468968)

[4.2.2 System context 16](#_Toc468468969)

[4.2.3 Sales Data Real Time Polling (POS – Staging) 17](#_Toc468468970)

[4.2.3.1 Process Flow 17](#_Toc468468971)

[4.2.3.2 Functional Requirement Details 20](#_Toc468468972)

[4.2.4 Sales Data Real Time Polling Flow (Staging - EDW) 23](#_Toc468468973)

[4.2.4.1 Processing Flow 23](#_Toc468468974)

[4.2.4.2 Functional Requirement Details 26](#_Toc468468975)

[4.2.5 Sales Data End-of-day (POS – Staging) 29](#_Toc468468976)

[4.2.5.1 Process Flow 29](#_Toc468468977)

[4.2.5.2 Functional Requirement Details 32](#_Toc468468978)

[4.2.6 Sales Data End-of-day (Staging – EDW) 34](#_Toc468468979)

[4.2.6.1 Process Flow 34](#_Toc468468980)

[4.2.6.2 Functional Requirement Details 37](#_Toc468468981)

[4.3 Pricing/Master Data 41](#_Toc468468982)

[4.3.1 Pricing/Master Data Overall Process Flow 41](#_Toc468468983)

[4.3.2 System context 42](#_Toc468468984)

[4.3.3 Pricing data generation and download to Staging 43](#_Toc468468985)

[4.3.3.1 Processing Flow 43](#_Toc468468986)

[4.3.3.2 Requirement Details 46](#_Toc468468987)

[4.3.4 Pricing/Master data distribution (Staging - POS) 47](#_Toc468468988)

[4.3.4.1 Process Flow 47](#_Toc468468989)

[4.3.5 Requirement Details 50](#_Toc468468990)

[4.4 Non-functional Requirement 52](#_Toc468468991)

[4.4.1 Requirement Details 52](#_Toc468468992)

[4.4.2 System Interface 57](#_Toc468468993)

[5 Sign Off 59](#_Toc468468994)

[6 Appendix A – Existing Polling Servers Overview 60](#_Toc468468995)

[7 Appendix B – Data Process Flowchart 60](#_Toc468468998)

# Background

## Document Purpose

The purpose of the Requirements Definition Document (RDD) is to describe the detailed business, functional and non-functional requirements for a project and its main aim is to provide business and functional context for the project and its objectives. It will provide the input for high level design activities and it will serve as the baseline against which detailed design documents and the implemented solution are assessed.

The Requirements Definition Document will provide a common understanding of all of the requirements for all project stakeholders, covering detailed business requirements, business process design, flow and business rules (aligned to the Maxim’s POS-oriented data processing including Sales, Master and Pricing data) as well as functional requirements, use cases and non-functional requirements.

The Requirements Definition Document is produced as part of the Business Case Development phase of Project Delivery Lifecycle.

## Document Scope

The scope of the Requirements Definition Document (RDD) is to describe the detailed business requirements, process design/flow, system context (outlining key applications used by the business unit/domain and the relationships that exist amongst them), functional requirements and uses cases (describing the functional behavior of the impacted applications) and non-functional requirements (describing the non-functional behavior of the supporting operations). The scope does not include technical designs or specifications of the impacted applications.

## Document Audience

The audience of the Requirements Definition document (RDD) is business users, project stakeholders, project teams, partners and suppliers.

## Terms & Abbreviation

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| ESB | Enterprise Service Bus |
| API | Application Programming Interface |
| EDW | Enterprise Data Warehouse |
| EOD | End of day |
| POS Client | One database owner on the sales side, provided by the POS machine vendor/manufacturer |
| DB | Database |
| POS | Point of Sales |

## Reference Materials

| Document Names |
| --- |
| Maxim POS Polling ESB Implementation Service Proposal EL-FY16-902-v3.docx |
| POS Polling User Requirement Confirmation-20161121-Discussion Note.xlsx |

# Executive Summary

The aim of the ESB project is to migrate three data processing flows in Maxim’s current enterprise architecture using database provided technologies (linked server, stored procedure) to the platform using enterprise services bean technology. In the new platform, two data processing flows will be provided:

* Sales data real time processing to EDW
* Sales data EOD processing to EDW

Depends on the configuration, Oracle ESB layer will use the JDBC to connect the database of the POS system in outlets by pre-configured connection information in order to collect sales data or updating pricing or master data. According to the existing polling logic found in IT51 server, when there is any connection error, Oracle ESB Layer will log the error and retry the data synchronization again. If the maximum retries reached, system will stop the synchronization for that particular POS system and generate alert for the synchronization failure. The POS connection information shall be retained as system configuration and can be flexibly maintained by Maxim IT.

For POS systems using DBF file as the exchange data media, the customized layer will directly read from / write to the DBF using the JDBC. The handling will be the same for both local database and DBF file.

The new ESB POS Polling can invoke several threads, which can be configured, to poll/push the data to the POS systems concurrently for maximizing performance.

# Business Context & Concept

## Business Rationale, Objectives & Considerations

### Objectives

| **#** | **Objective** | **Description** |
| --- | --- | --- |
| 1 | Bridge the POS clients and EDW to transfer real time sales data through Service Bus | Using Service Bus and an enterprise application layer to take over the responsibility & functionality of real time polling from Maxim’s POS clients which are currently done by SQL Server linked server and SQL agent jobs |
| 2 | Bridge the POS clients and EDW to transfer EOD sales data through Service Bus | Refer to #1 |
| 3 | Enable the pricing/master data distribution to POS clients through Service Bus | Refer to #1 |

**1 Objectives**

### Expected Business Benefits

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Outcome** | **Performance Indicator** | **Measure** | **Baseline** | **Target Level** | **Accountability** |
| 1 | Exception Control | Information of the exception logs & exception report | Flexibility in exception handling and job control | Existing exception handling | Moderate | High |
| 2 | POS client Maintenance | Ease of add or remove, pause or resume a POS client. | Ease of job control and manipulation | Existing SQL agent job control | Moderate | High |

2 Expected Business Benefits

### Considerations

#### Assumptions

| # | Assumption | Description | Owner |
| --- | --- | --- | --- |
| 1 | JDBC connection to POS clients | The service bus application layer will maintain JDBC connection to collect data from POS clients DB | Carl |
| 2 | Sale & sale EDO data pushing to EDW | Service bus will write sales and sales EOD data into corresponding staging table in EDW then EDW will trigger conversion process against these data to formalize them into other table. The staging tables’ schema shall be the same as those in POS clients/Service Bus staging DB because Service bus will minimize the data transformation | Polly |
| 3 | Covert log for EOD process | The convert log record is the only reference from POS which allow service bus to acknowledge the EOD processing | Carl |
| 4 | History Records | Assume the history tables are ready for download in POS client once the convert log is written | Carl |
| 5 | Exception handling & report | Oracle ESB Exception Report shows only the exception for those ESB enabled POS system. For those non-ESB enabled POs system, error message should refer to the existing POS Polling process. The application layer handles most of the exception handling according to user requirements. Refer to latter sessions details | Carl |
| 6 | Real time sales data check sum mechanism – MITPOS | The sales data from MITPOS could be considered as completed sales order data. | Carl |
| 7 | Void order data – MITPOS | The voided sales order would become another reverse order records from POS client to service bus | Carl |
| 8 | Direct data-to-date EOD process | EOD processing, EOD data will be synchronized to ESB Application DB and then do a direct DB-to-DB copy to EDW staging table. System only ensure the total count and the amount is matched between these two systems. There is NO logic for the copying process. | Wing |
| 9 | Pricing/Master data update to POS client | The pricing/master update should be maintained the grouped data together in one transaction. The grouping information could be found in “Poll\_schema\_info” | Carl |
| 10 | Pricing/Master data update to POS clients | Assume the pricing/master data has the primary key, the processing will update/merge the data by the reference key to avoid override. | Carl |

#### Constraints

| **#** | **Constraint** | **Description** | **Owner** |
| --- | --- | --- | --- |
| 1 | Real time sales data check sum mechanism - POINTSOFT | Because the records from POINTSOFT POS client would alter (increase and decrease) when the order is changed by time, hence, the checking on the completeness of a sales order would be limited to   1. When no discount - total amount of the order item equals to the order payment amount 2. When having discount – Total amount minus discount amount should equal to payment amount | Carl |
| 2 | EOD data process no compromise | EOD data is full data set transfer from history tables in POS clients to EDW tables, and service bus process is not able to do a proper check sum. Service bus could only ensure a total record count. The check sum can’t only be done by EDW conversion process | Wing |
| 3 | Real time Sales data polling | The real time sales data polling process is required to run 24x7.  DB connection will be initiated in a timed interval even though there’s a possibility that the POS client is offline. | Carl |
| 4 | POS client pricing/master data update mechanism | No need to delete the record in the client POS system. The update strategy is. If record exists in the client local DB, update record according to the key (NOT delete and insert). If record not exist, just insert NEW records. if DELETE action, system only logical mark delete the record but NOT physical delete record | Carl |
| 5 | Pricing/Master data update in Pricing server | Because in current system of pricing server, it requires triggering the stored procedures to generate pricing data, so that service bus need to implement a DB adapter to trigger this process by time interval rather than use a DB adapter to monitor the pricing/master data’s delta change. | Wing |

#### Dependencies

| **#** | **Dependency** | **Description** | **Owner** |
| --- | --- | --- | --- |
| - | - | - | - |

#### Issues

| **#** | **Issue** | **Description** |
| --- | --- | --- |
| - | - | - |

#### Risks

| # | Risk | Likelihood | Owner | Severity | Potential Impact/Mitigation Strategy |
| --- | --- | --- | --- | --- | --- |
| - | - | - | - | - | - |

## Value Propositions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Value Proposition** | **Type** | **Area of Benefit** |
| - | - | - | - |

# Business Process

## Business Process Scope

Process Area

|  |  |
| --- | --- |
| Name | Description |
| Sales data real time exchange | In Maxim’s daily operation, the sales data in POS clients, will be collected by 2 parties of data consumers in real time.   1. Sales operation team – for data backup, auditing and other business process. 2. EDW team – use partial data for data conversion and analysis purpose.   The sales data shall be collected from 400+ POS clients which are running upon 3 different POS solution vendors, with different database models and interface methods. The service bus between the data providers (POS clients) and the data consumers is expected not only to fulfil the purpose of data polling and pushing but also to bridge these differences. |
| Sales data end-of-data processing | The POS client shall conduct end-of-day process before the cut of time (next day 4:00am) in each opening day. The end-of-day process shares the similar mechanism of real time process but requires service bus to monitor the client’s EOD flag which indicates that client has finished local data processing and ready for polling.  In the EOD processing, the service bus shall provide a check record count for the client by validating the total records received in real time within that day, so as to ensure the |
| Master/Pricing data processing | According to Maxim, the Master data and Pricing data shall be generated together in the backend, and distributed to POS clients with time interval batch jobs. These data has dependencies between some business related entities and requires their commitment control to be done within one single transaction against one POS client. |

Process Group

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Business Owner | Business Area / Domain |
| Sales Data Download to Staging DB | POS and EDW sales data Download to Service Bus staging DB (Real time & EOD) | Carl | POS |
| Send POS sales data from Staging DB to EDW | Service Bus send sales data from staging to EDW  (Real time & EOD) | Polly | EDW |
| Pricing/Master Data Processing & Download | POS and Pricing Server data exchange, and polling schema definition | Wing | Infra |
| Pricing/Master Data Distribution | Service Bus application distribute/send pricing/master date from staging DB to all POS clients | Carl | POS |

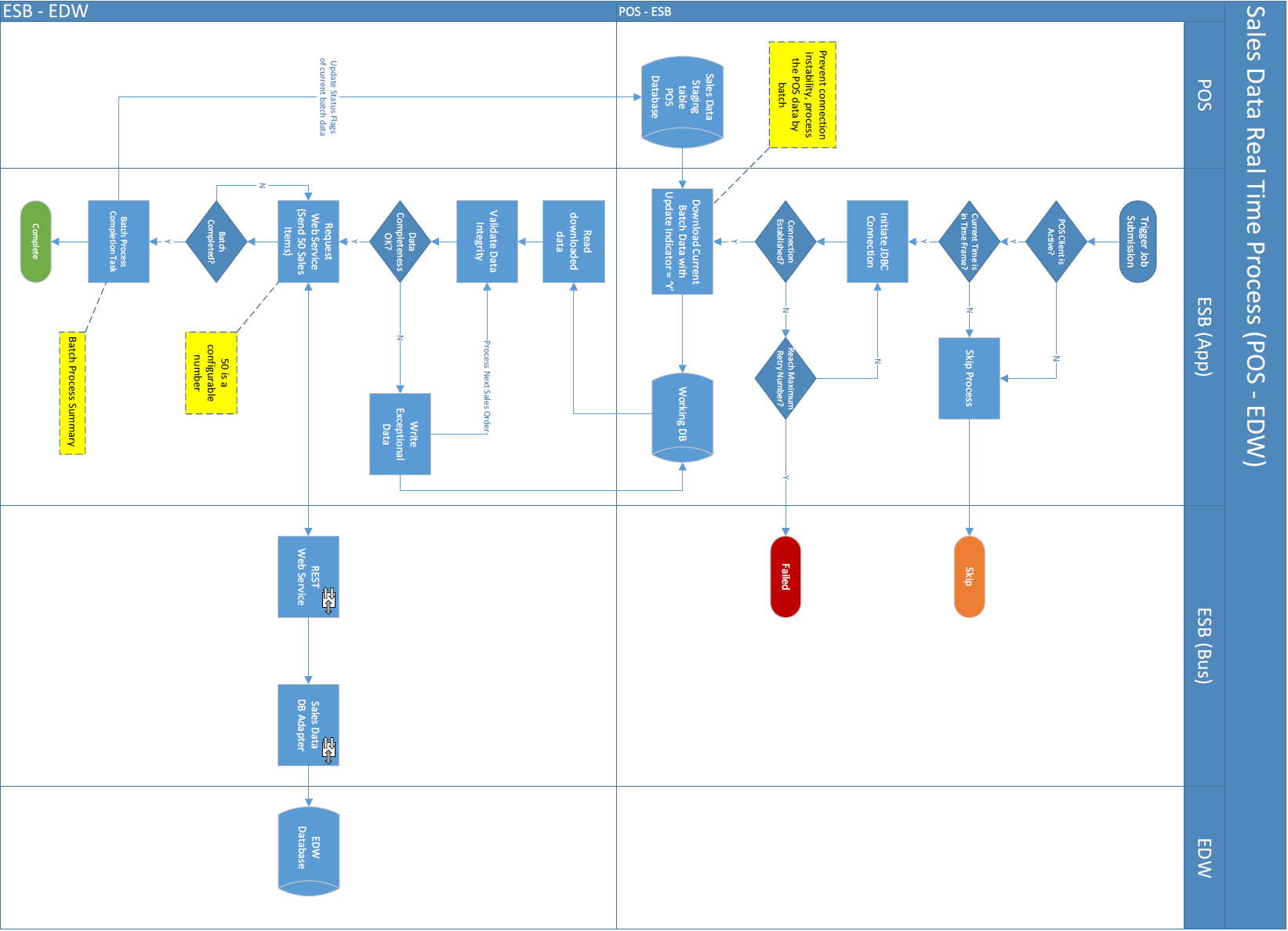
Business Process

|  |  |  |
| --- | --- | --- |
| Name | Description | Objective |
| Sales Data Real time Exchange | In 7x24 time frame, collecting the data from POS clients and put them into EDW | To fulfil the purpose that the sales data in POS clients must be synchronized to Maxim Headquarters’ EDW in real time |
| Sales Data End-of-day Process | Running one time each time (maximally) to copy data from POS client to EDW | To transfer EOD data from each POS clients to EDW tables by on-demand triggered. |
| Master/pricing data generation and update to POS client | When master/pricing data are updated, push the update to corresponding POS client. | To copy data from pricing/master data DB and distribute them to POS client according to the pricing group setting. |

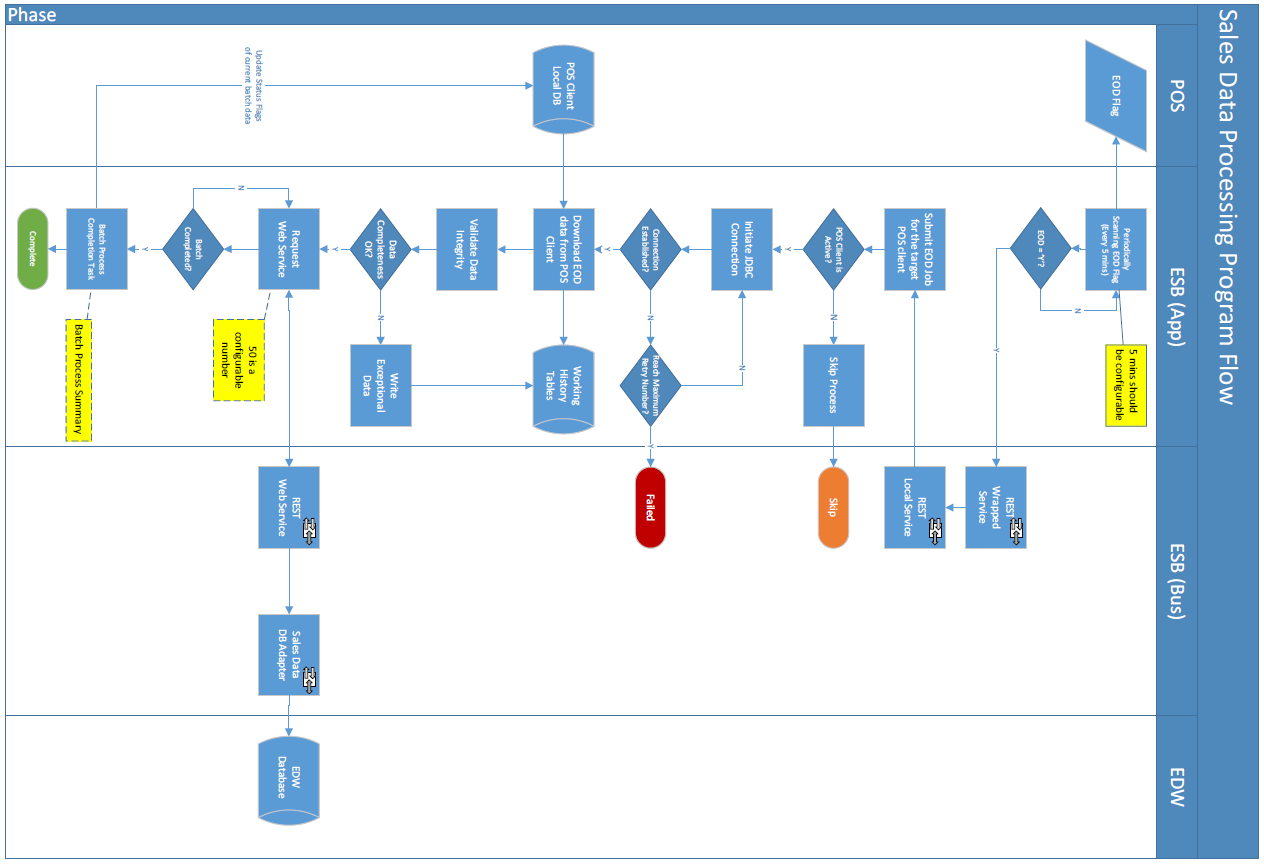
## Sales Data

### Sales Data Overall Process Flow

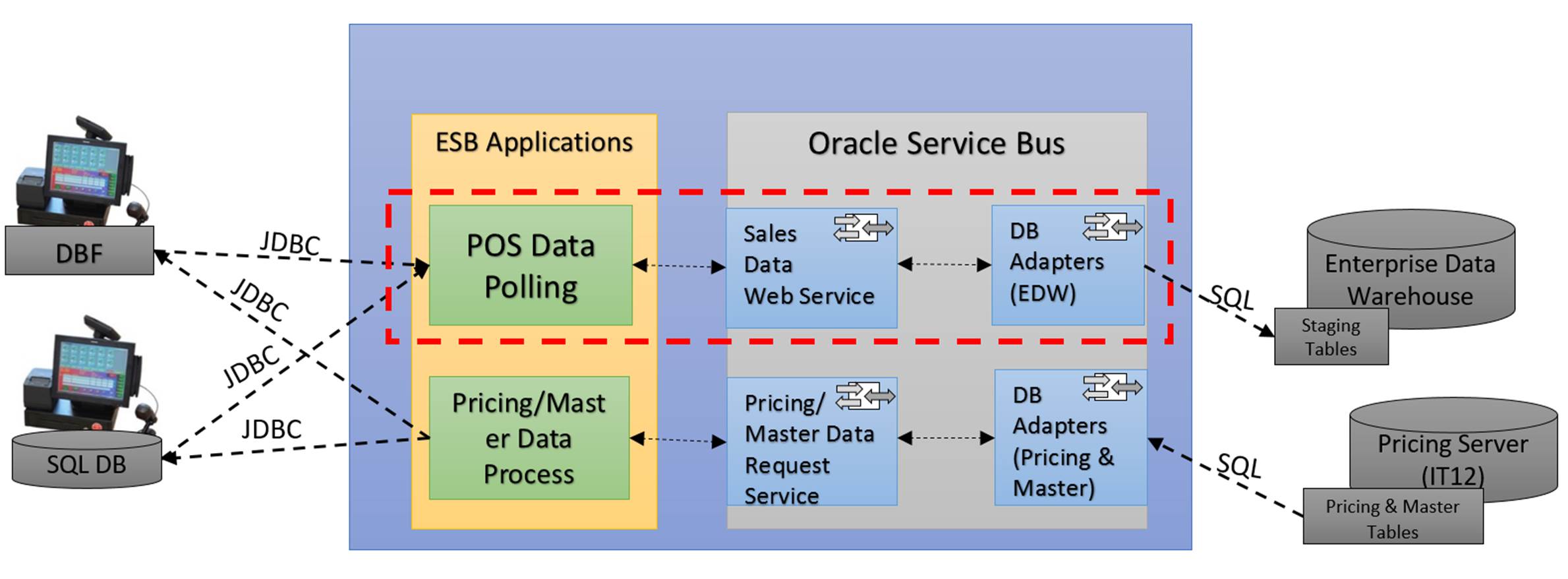
**Sales Data Real Time Processing**



**Sales Data End-of-day Processing**

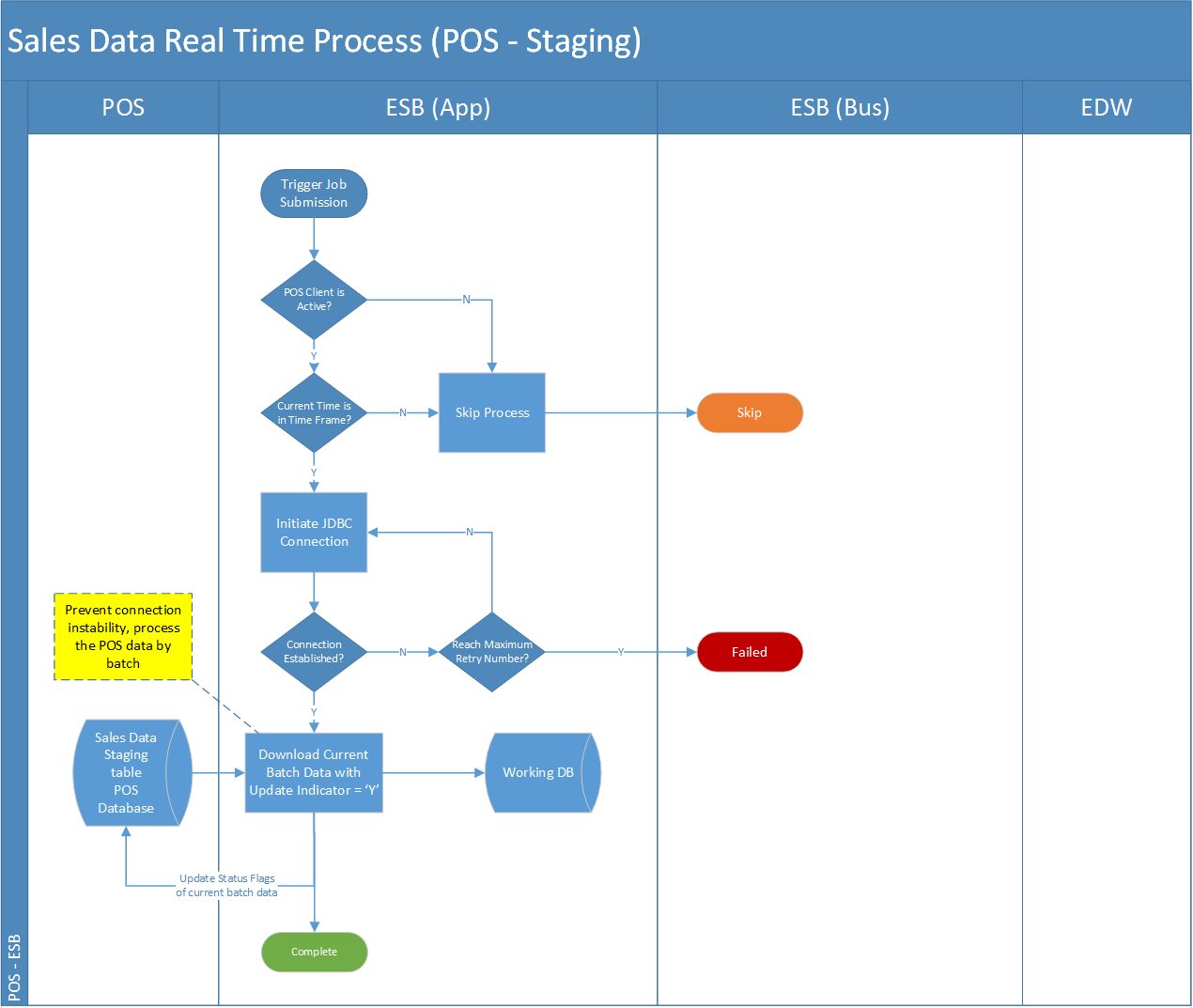


### System context



### Sales Data Real Time Polling (POS – Staging)

#### Process Flow



Task

| Activity | Description | Type | Process Group - Owner |
| --- | --- | --- | --- |
| ESB triggers the batch processing again the POS client | 1. The POS polling job pool is controlled by a centralized job controller which runs in a timed interval 2. The time trigger shall submit multiple jobs in the pool, which sequentially process the POS client’s sales data polling in round robin rules according to the list of POS client defined (similar to the poll\_scheme\_control\_table and the branch data in IT50.maxim.dbo.branch) 3. The job controller shall justify the job submission according to the setting from the schema control table   Please mentioned parallel processing of Sales/EOD, Pricing and Master for POS client synchronization. | Process | Sales Data Download to Staging DB - Carl |
| Download sales data from POS clients to service bus working database | 1. For MS SQL basis POS client, e.g. MITPOS, The POS client polling job will initiate the JDBC connection to the target data source, and download polling data set which is “pending” for poll in POS client’s DB. 2. If the data source is a DBF file, the job will scan the target directory configured in the polling schema info table, and use the DBF data to override the data in the staging tables.   *Note: Assumption has been made that POINTSOFT POS client should use FTP transfer the DBF file to ESB application server’s local directory.* | Input | Sales Data Download to Staging DB - Carl |

Role

| Role | Description | Type |
| --- | --- | --- |
| System Administrator | The user role has the right to access the admin function of the system, e.g. job control table configuration, data source schema control configuration, parameter configuration, etc. | Admin |
| System Operator | The user role has to right to conduct job re-run, job logs and dashboard. And demander of daily summary report | Operator |
| POS Support | The user role has to right to conduct job re-run, job logs the responsibility to receive POS related alert email. | Support |

Application

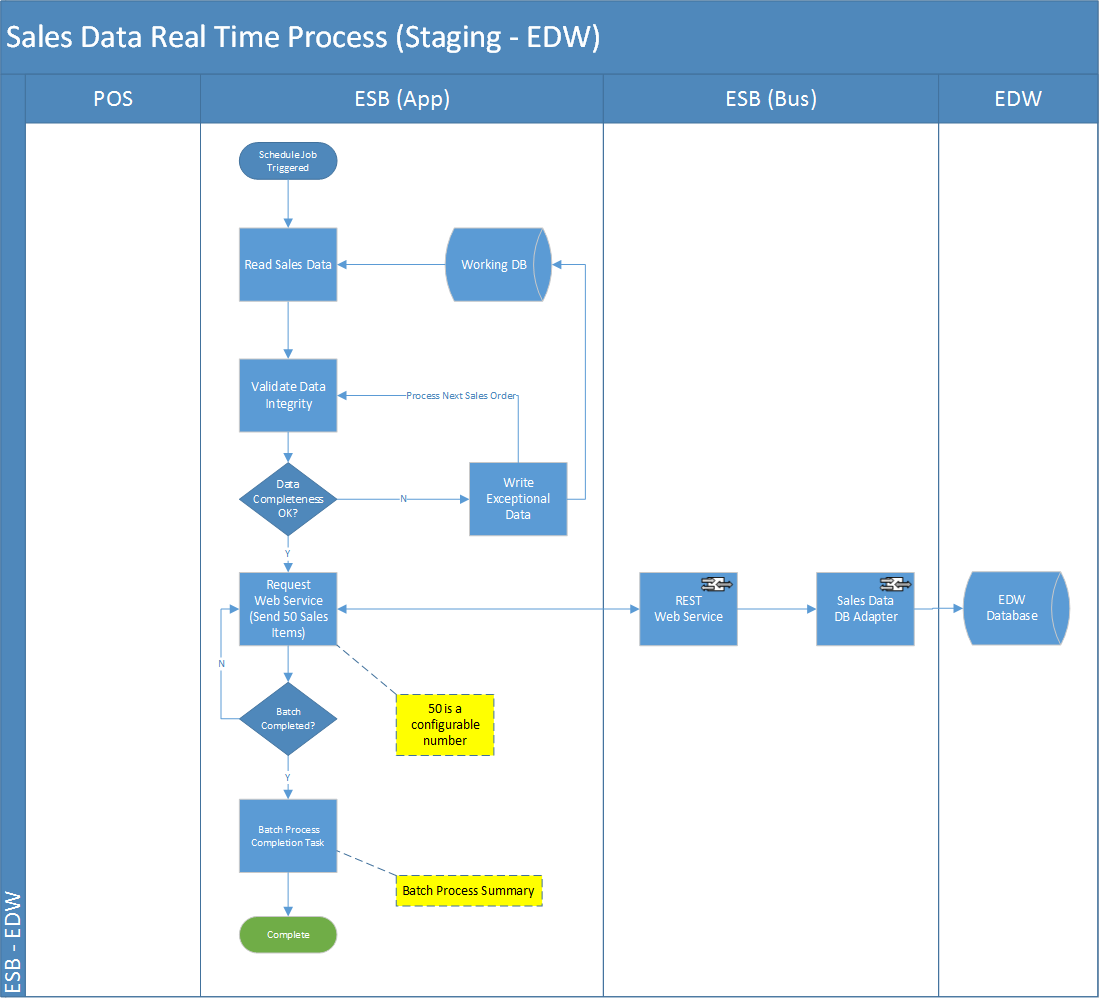
| Application | Description |
| --- | --- |
| Polling Gateway | The application module responsible of downloading data from POS client. |
| Sending Gateway | The application module to carry out data pushing to service bus interfaces. |
| ESB Interfaces | The interface for data pushing from service bus application layer staging DB to EDW. |

#### Functional Requirement Details

| **ID** | **Name** | **Description** | **Priority** | **Owner** | **Logic** | **Parent** |
| --- | --- | --- | --- | --- | --- | --- |
| ESB-SAL-001 | Sales Data Real time polling to Staging DB | The sales data in POS clients, will be collected to Service Bus Staging DB in real time. | Must Have | Carl | - | --- |
| ESB-SAL-001-01 | Data staging to service bus working DB | In real time, the service bus shall trigger schedule jobs to download data from POS clients’ database. | Must Have | Carl | The schedule runs in a time interval no longer than 15mins. (current SQL Agent schedule job time)  The running period of the schedule shall be 24x7 polling. | ESB-SAL-001 |
| ESB-SAL-01-02 | Data staging to service bus working DB | The schedule job shall handle 2 kinds of data source connection according to different POS machine vendors. | Must Have | Carl | 1. POINTSOFT – DBF file   The DBF file contains the full set of data of target tables from POS client’s DB.   1. MITPOS – SQL Server database connection   For the database connection, the schedule job shall download the delta data which is indicated by the flags in local tables of POS clients. | ESB-SAL-001 |
| ESB-SAL-01-03 | Data staging to service bus working DB | The service bus polling application will take care of the sales order transactions according to different POS client.   * MITPOS | Must Have | Carl | MITPOS client will commit a “full sales order transaction” as a whole, which means ORDER, TRANS and PAY are committed together when a purchase order is completed. When there’s void order, MITPOS client will generate voided transaction for the corresponding records. | ESB-SAL-001 |
| ESB-SAL-01-04 | Data staging to service bus working DB | The service bus polling application will take care of the sales order transactions according to different POS client.   * POINTSOFT | Must Have | Carl | POINTSOFT POS will commit by tables instead of "Sales", partial record will be found.  For Sales Polling, only "whole" sales transactions should be sent to EDW for Sales data Polling. Partial sales records (e.g. POINTSOFT POS) should NOT be sent to EDW. The data from POINTSOFT client will being staged in service bus working DB until the data become consistent. | ESB-SAL-001 |
| ESB-SAL-01-05 | Update POS Client’s Polling Flag after | Update POS Client’s Polling Flag after the download of data is successful. | Must Have | Carl | After downloading the sales data from POS client, if the transfer is successful, the processing job update the polling flag status back to POS client indicating the batch of data has already been polled to service bus staging DB upon the job completion. | ESB-SAL-001 |
| ESB-SAL-01-06 | Job Completion | The job completion shall log information of the job processing in DB for real time processing. | Must Have | Carl | 1. Job run status 2. Number of records processed 3. Job name, e.g. POS – Staging 4. Job ID 5. Job run error stage (if any) 6. Error Category, e.g. DB, Network and Other, etc. 7. Job start/end time (regarded as Coping to Staging start/end time) | ESB-SAL-001 |

### Sales Data Real Time Polling Flow (Staging - EDW)

#### Processing Flow



Task

| Activity | Description | Type | Process Group - Owner |
| --- | --- | --- | --- |
| Read sales data from Staging DB | 1. Another job will be submitted to process the real time sales data in the staging DB of one branch.   -- Another Job: Staging ESB to EDW | Process | Send POS sales data from Staging DB to EDW – Polly |
| Validate the data integrity | 1. The job will continue to validate (refer to requirement details of Sales Data real time processing in below section <<- Add section back number) the polled sales data’s integrity in the staging table of current batch. If the whole batch data’s completeness is alright, the job will move to next steps. | Process | Send POS sales data from Staging DB to EDW – Polly |
| Send the sales data to EDW | 1. The job will send the data to EDW through the pre-configured interface in the service bus, and the interface shall fulfill below function.    1. Transfer sales data apple-to-apple from staging DB to EDW    2. Response a return code indicating the data transfer is success or not. The response will contain the records ID with corresponding return code. 2. If the transfer is successful, the response result will be updated into Service Bus staging database, and lately revert back to Staging DB upon the job completion. | Output | Send POS sales data from Staging DB to EDW - Polly |
| Job Completion | 1. Update the job log to reflect the summary of the whole real time batch including:    1. Job run status    2. Number of records processed    3. Job name, e.g. Staging – EDW.    4. Job ID    5. Job run error stage (if any)    6. Error Category, e.g. DB, Network and Other, etc.    7. Job start/end time (regarded as Staging to EDW start/end time) | Process | Sales Data Download to Staging DB – Carl  Send POS sales data from Staging DB to EDW - Polly |

Measure

| Measure | Description |
| --- | --- |
| - | - |

Role

| Role | Description | Type |
| --- | --- | --- |
| System Administrator | The user role has the right to access the admin function of the system, e.g. job control table configuration, data source schema control configuration, parameter configuration, etc. | Admin |
| System Operator | The user role has to right to conduct job re-run, job logs and dashboard. And demander of daily summary report. | Operator |
| Service Bus Support Team | The user role has to right to conduct job re-run, job logs the responsibility to receive EDW related alert email. | Support |

Application

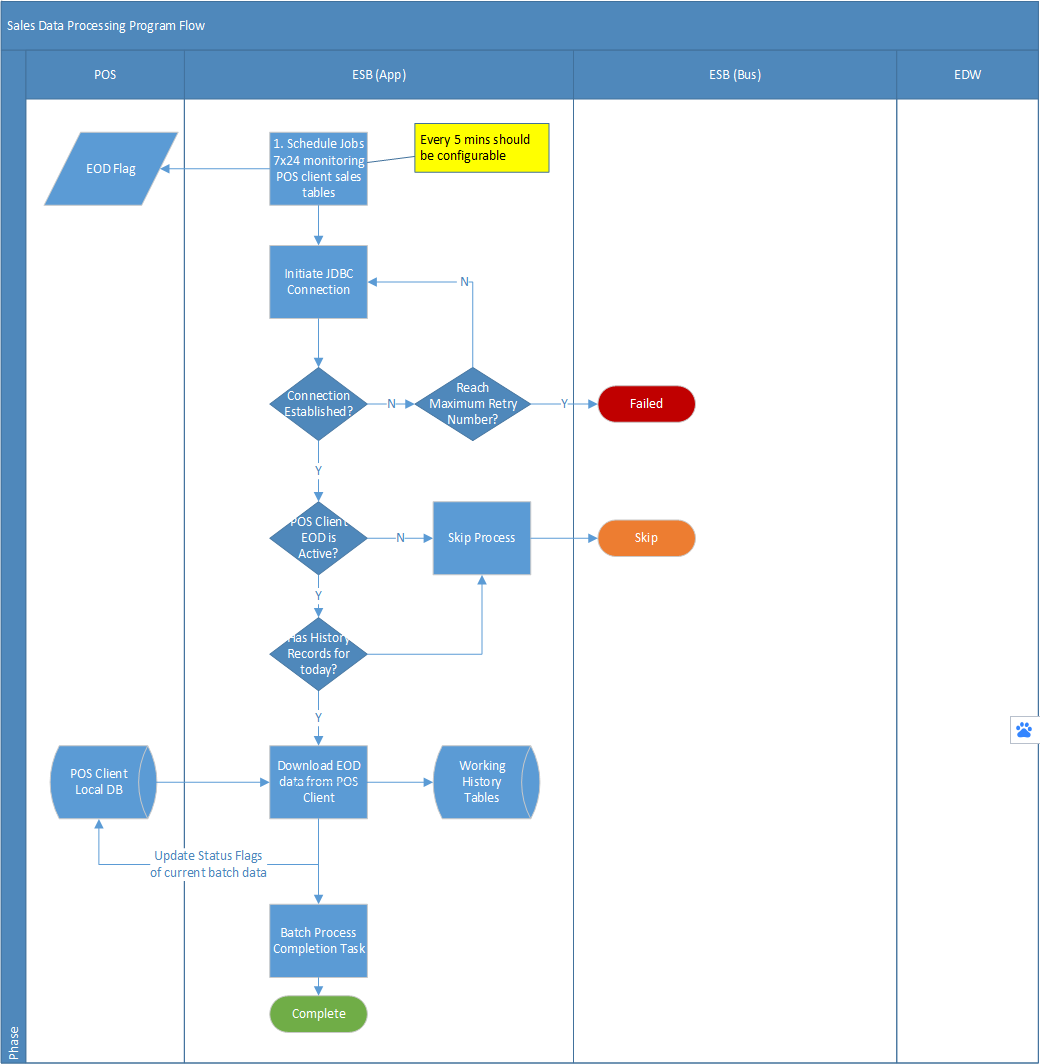
| Application | Description |
| --- | --- |
| Polling Gateway | The application module responsible of downloading data from POS client. |
| Sending Gateway | The application module to carry out data pushing to service bus interfaces. |
| ESB Interfaces | The interface for data pushing from service bus application layer staging DB to EDW. |

#### Functional Requirement Details

| **ID** | **Name** | **Description** | **Priority** | **Owner** | **Logic** | **Parent** |
| --- | --- | --- | --- | --- | --- | --- |
| ESB-SAL-02 | Data processing to EDW | - | - | - | - | - |
| ESB-SAL-02-01 | Data processing to EDW | The step 2 processing job reads sales data from staging DB. Firstly it shall check the Data integrity for data from both MIDPOS and POINTSOFT. | Must Have | Polly | According to #ESB-SAL-01-03 & # ESB-SAL-01-04  a) orders, trans, order\_pay, coupon, double discount. Including tables list, where tables are mandatory or not.  b) Tables are linked by  - Business Date  - Branch Code  - Sales Order ID.  c) All sales tables can be linked up by the above key.  d) In an complete order, the order items total amount must equal to the payment amount | ESB-SAL-02 |
| ESB-SAL-02-02 | Data processing to EDW | Data processing job will focus on real time sales data related target table. | Must Have | Polly | * Check\_log * Coupon sales * Orders * Orders pay * Redeem coupon * Supp * Trans * Transaction ecard * Transaction modifier * Order extra info |  |
| ESB-SAL-02-03 | Data processing to EDW | After verifying the sales data is completed. Send the completed data to EDW. | Must Have | Polly | The service bus application will use above keys to check the integrity of data. When the data in 3 tables are confirmed completed, the service bus application will send the full order data to EDW. | ESB-SAL-02 |
| ESB-SAL-02-04 | Data processing to EDW | Real time sales data polling exceptional data handling. | Must Have | Polly | Theoretically the data from POS client shall not contain the in-consistent data to service bus. However, as #06, because the data updating logic of POINTSOFT is full table override, the service bus application will wait until the data of a full purchase order to complete, and send to EDW.  There will be an indicator in the staging database reflecting the records are being sent or not. Until the polling cut-off time, the system shall send the alert email with accumulated exceptional data to support team. | ESB-SAL-02 |
| ESB-SAL-02-05 | Data processing to EDW | Sales data sending, by block update and return code. With the return code of the service call, the process will mark the block of records the completion of the sending to EDW.  When return value is a failure code, the process applies a retry mechanism with the retry limit. If the number of retry exceeds the limit, it shall skip to next block. | Must Have | Polly | The service bus has the limitation of network, the number of records transferred shall be done block by block (e.g. 50). This number shall be designed as a configurable item in the system.  When the service response success return code, update to the staging tables indicating the corresponding records are synchronized. | ESB-SAL-02 |
| ESB-SAL-02-06 | Job Completion | The job completion shall log information of the job processing in DB for real time processing. | Must Have | Polly | 1. Job run status 2. Number of records processed 3. Job name, e.g. Staging - EDW 4. Job ID 5. Job run error stage (if any) 6. Error Category, e.g. DB, Network and Other, etc. 7. Job start/end time (regarded as Coping to EDW start/end time) | ESB-SAL-002 |

### Sales Data End-of-day (POS – Staging)

#### Process Flow



Task

| Activity | Description | Type | Process Group/Owner |
| --- | --- | --- | --- |
| Monitor POS client’s completion of its EOD process. | 1. The enterprise service bus process shall monitor the update of the “hist\_possystem” which was written in POS clients. 2. On retrieved the triggering value of this record, the EOD job controller will fire a job to start the EOD process towards this data source (MITPOS). | Process | Sales Data Download to Staging DB - Carl |
| Download EOD sales data from POS clients to service bus working database | 1. The EOD processing job downloads full set of data from POS client’s history tables, and stage them in service bus DB. | Input | Sales Data Download to Staging DB - Carl |
| Validate the data integrity | 1. After successfully written into staging DB, The processing job will conduct a check sum upon the data set with below criteria: 2. Number of record count received 3. Total Amount check sum 4. Creation Date Time of Sales records (HIST\_ORDER) is before or after last day’s cut-off time, and adjust it business date according to the cut-off logic (refer to requirement details for more info) | Process | Sales Data Download to Staging DB - Carl |
| Job Completion | 1. Update the job log to reflect the summary of the whole EOD process including:    1. Job run status    2. Number of records processed    3. Job name, e.g. POS - Staging    4. Job ID    5. Job run error stage (if any)    6. Error Category, e.g. DB, Network and Other, etc.    7. Job start/end time (regarded as Coping to EDW start/end time) | Process | Sales Data Download to Staging DB - Carl |

Role

| Role | Description | Type |
| --- | --- | --- |
| System Administrator | The user role has the right to access the admin function of the system, e.g. job control table configuration, data source schema control configuration, parameter configuration, etc. | Admin |
| System Operator | The user role has to right to conduct job re-run, job logs and dashboard. And demander of daily summary report. | Operator |
| POS Support | The user role has to right to conduct job re-run, job logs the responsibility to receive POS related alert email. | Support |

Application

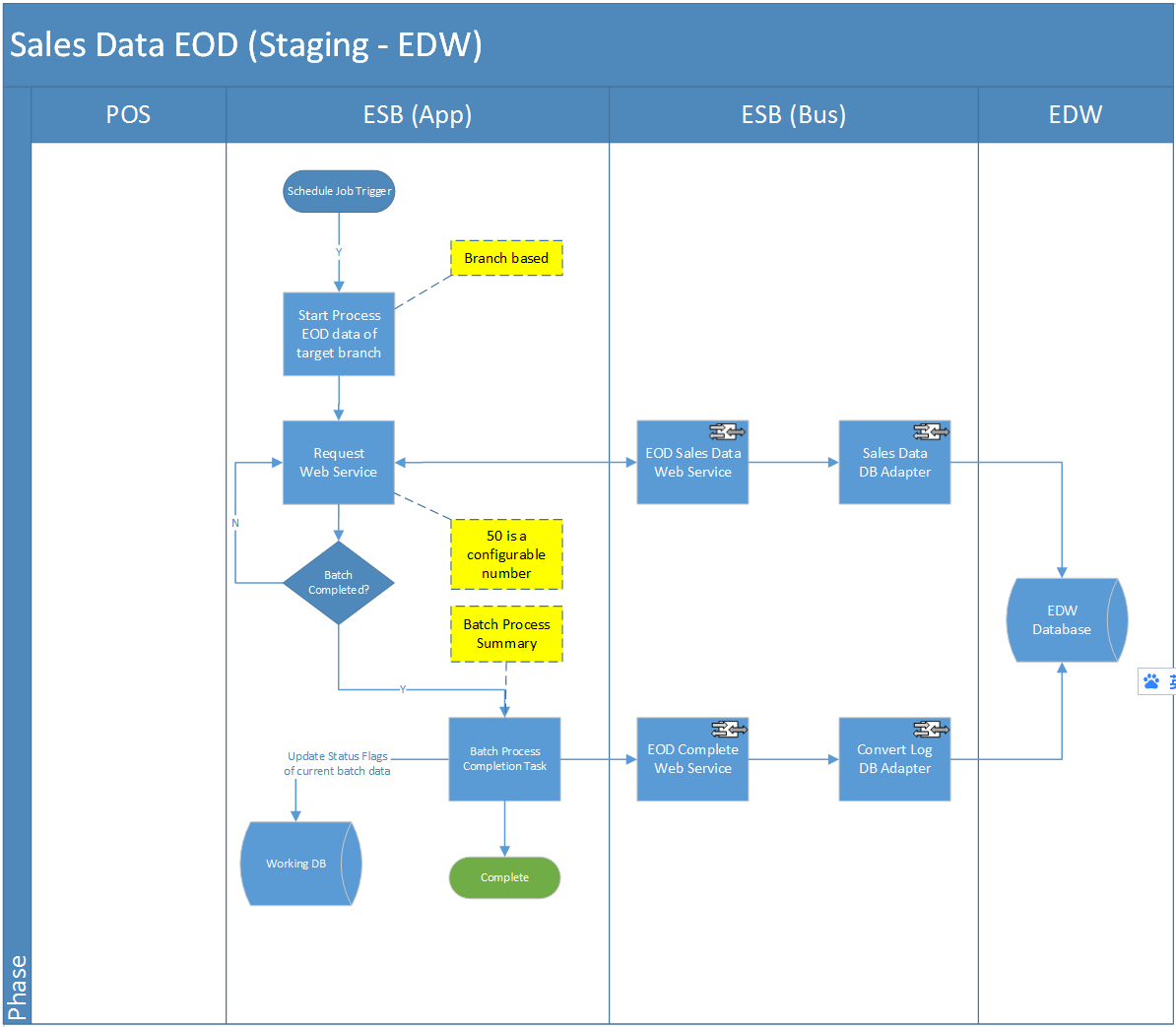
| Application | Description |
| --- | --- |
| Polling Gateway | The application module responsible of downloading data from POS client. |
| Sending Gateway | The application module to carry out data pushing to service bus interfaces. |
| ESB Interfaces | The interface for data pushing from service bus application layer staging DB to EDW. |

#### Functional Requirement Details

| **ID** | **Name** | **Description** | **Priority** | **Owner** | **Logic** | **Parent** |
| --- | --- | --- | --- | --- | --- | --- |
| ESB-SAL-REQ-03 | End of day data processing | - | - | - | - | - |
| ESB-SAL-REQ -03-01 | POS client triggering | After the POS client completes its EOD processing, it insert a new record into ”hist\_possystem“ table indicating it’s ready for the EOD polling. | Must Have | Carl, | The convert log indicates the EOD process’s completion on POS client side and ready for polling server’s action.  Service Bus application shall take it as a reference signal as the EOD processing start. | ESB-SAL-REQ-03 |
| ESB-SAL-REQ -03-02 | POS Client EOD Monitoring | The service bus shall run an application monitoring the local end-of-day indicator in the POS database. | Must Have | Carl | Because current POS clients do not have the capability of web service call, the service bus is not able to trigger the EOD process passively. So, it requires the service bus application will be able to running a job keep monitoring the local EOD flag. When discovering the flag has changed to the EOD value, submit a job to start the EOD process for that POS client. | ESB-SAL-REQ-03 |
| ESB-SAL-REQ -03-03 | POS Client EOD Data Process | The service bus initiate a JDBC connection to download all EOD history tables from POS client to its staging DB. | Must Have | Carl | To avoid holding the DB connection too long, or interruption of processing if confronted network issue, the EOD process also downloads the full set of history data into staging DB first. | ESB-SAL-REQ-03 |
| ESB-SAL-REQ -03-04 | POS Client EOD Data Process | Update data synchronization status in Staging database and POS client. | Must Have | Carl | After the history data is successfully downloaded to Staging, the result shall revert back to the POS clients. | ESB-SAL-REQ-03 |
| ESB-SAL-REQ -03-05 | EOD process job summary | The end of the job also log down the demanded information for user to collect details of data of the job running progress & result. | Must Have | Carl | 1. Update the job log to reflect the summary of the whole EOD process including:    1. Job run status    2. Number of records processed    3. Job name, e.g. POS - Staging    4. Job ID    5. Job run error stage (if any)    6. Error Category, e.g. DB, Network and Other, etc. 2. Job start/end time (regarded as Coping to EDW start/end time) | ESB-SAL-REQ-03 |

### Sales Data End-of-day (Staging – EDW)

#### Process Flow



Task

| Activity | Description | Type | Process Group/Owner |
| --- | --- | --- | --- |
| Monitor POS client’s completion of its EOD process. | 1. Upon the completion of downloading EOD data of the target branch, a step 2 process is triggered to read those data (same branch). | Process | Send POS sales data from Staging DB to EDW - Polly |
| Send the sales data to EDW | 1. Send the EOD data to EDW with block by block transaction commitment through the Service Bus. | Output | Send POS sales data from Staging DB to EDW - Polly |
| Revert the Sending results back to POS client | 1. If the transfer is successful, the response result will be updated into Service Bus staging database, and lately revert back to Staging upon service call response. | Output | Send POS sales data from Staging DB to EDW - Polly |
| Validation on Send data | 1. Upon the finish of the process, triggers a service to validate the total count & total amount of the transferred data on EDW side. If correct then call another service to write a convert log to EDW indicating the EOD process for that branch is completed. | Output | Send POS sales data from Staging DB to EDW - Polly |
| Job Completion | 1. Update the job log to reflect the summary of the whole EOD process including    1. Job run status    2. Number of records processed    3. Job name, e.g. EOD – Staging – EDW    4. Job ID    5. Job run error stage (if any)    6. Error Category, e.g. DB, Network and Other, etc.    7. Job start/end time (regarded as Coping to EDW start/end time) | Process | Send POS sales data from Staging DB to EDW - Polly |

Role

| Role | Description | Type |
| --- | --- | --- |
| System Administrator | The user role has the right to access the admin function of the system, e.g. job control table configuration, data source schema control configuration, parameter configuration, etc. | Admin |
| System Operator | The user role has to right to conduct job re-run, job logs and dashboard. And demander of daily summary report | Operator |
| EDW Support | The user role has to right to conduct job re-run, job logs the responsibility to receive EDW related alert email | Support |

Application

| Application | Description |
| --- | --- |
| Polling Gateway | The application module responsible of downloading data from POS client. |
| Sending Gateway | The application module to carry out data pushing to service bus interfaces |
| ESB Interfaces | The interface for data pushing from service bus application layer staging DB to EDW |

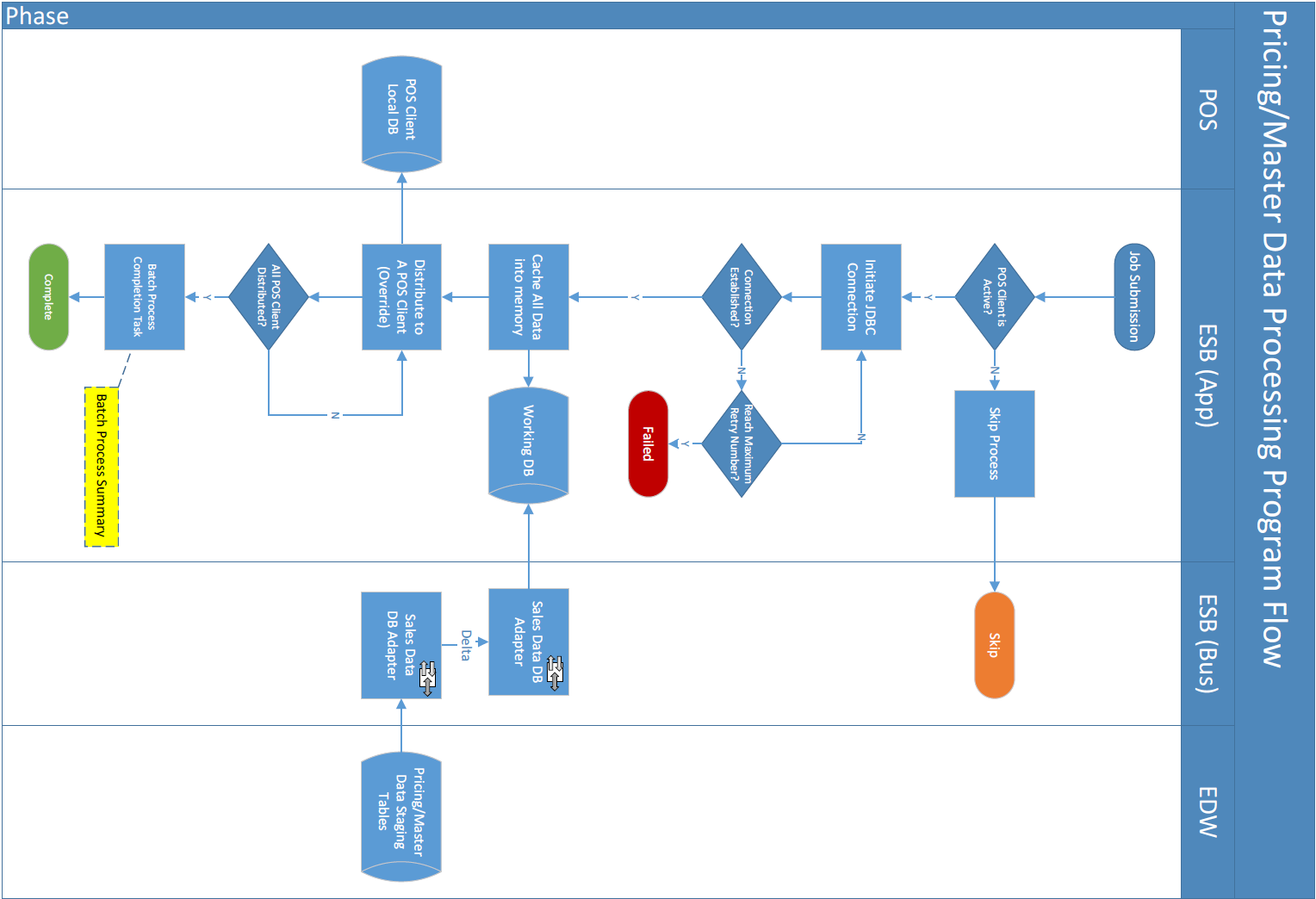
#### Functional Requirement Details

| **ID** | **Name** | **Description** | **Priority** | **Owner** | **Logic** | **Parent** |
| --- | --- | --- | --- | --- | --- | --- |
| ESB-SAL-REQ-04 | End of day data processing | - | - | - | - | - |
| ESB-SAL-REQ -04-01 | POS Client EOD Data Process | The EOD data will directly transfer the full data set to EDW | Must Have | Polly | EOD data will be synchronized under a direct DB-to-DB copy to EDW staging table. System only ensure the total count and the amount is matched between these two systems. There is NO logic for the copying process.  Since the EOD data does not have clear criteria of check sum and consistency, service bus will assumed the data is fully completed by POS client, and copy to EDW direct without integrity checking. | ESB-SAL-REQ-04 |
| ESB-SAL-REQ -04-02 | POS Client EOD Data Process | Service bus EOD process will also need to distinct the data’s business date when copying the data from POS client to staging database.  Because sometimes the POS client will accumulate multiple day’s data in one EOD process, the service bus EOD job shall identify this kind of data which belongs to different business date. Hence, the process shall justify the raw data’s creation date against the cut-off time (4a.m.). | Must Have | Polly | 1. When record creation time < last cut-off time, then business date = the day before 2. When record creation time > last cut-off time, then business date = the day after   For example,  EOD records in HIST\_ORDER  Business Date, Create Date Time  25-Nov-2016, 25-Nov-2016 09:12 🡺 No change  25-Nov-2016, 25-Nov-2016 04:11 🡺 No change  ------ Cut off Time (04:00) ---------  25-Nov-2016, 25-Nov-2016 02:12 🡺 Change to 24-Nov-2016  25-Nov-2016, 24-Nov-2016 06:33 🡺 Change to 25-Nov-2016  ------ Cut off Time (04:00) ---------  25-Nov-2016, 24-Nov-2016 03:33 🡺 Change to 23-Nov-2016 | ESB-SAL-REQ-04 |
| ESB-SAL-REQ -04-03 | POS Client EOD Data Process | EOD data process exception handling shall accumulate the exception data in a table and pending for operation team’s manual involvement. | Must Have | Polly | When the EOD process encounter unexpected exception, the process will skip these data, putting them into exceptional data table and wait for manual retry. Possible exception will be as below:   1. EDW database connection lost (this can be retry with retry limit) 2. Data cannot be insert into EDW tables (regarded as service call failure) 3. Timeout error, the insert process into EDW makes the application wait exceeds a time limit (60s), this can be retry with retry limit | ESB-SAL-REQ-04 |
| ESB-SAL-REQ -04-04 | POS Client EOD Data Process | Update data synchronization status in Staging database and POS client | Must Have | Polly | After the EOD data is successfully sent to EDW, the sending result shall revert back to service bus application’s staging database, and correspondingly update the POS clients. | ESB-SAL-REQ-04 |
| ESB-SAL-REQ -04-05 | EOD process job summary | The end of the job also log down the demanded information for user to collect details of data of the job running progress & result | Must Have | Polly | 1. Update the job log to reflect the summary of the whole EOD process including:    1. Job run status    2. Number of records processed    3. Job name, e.g. EOD – Staging – EDW    4. Job ID    5. Job run error stage (if any)    6. Error Category, e.g. DB, Network and Other, etc. 2. Job start/end time (regarded as Coping to EDW start/end time) | ESB-SAL-REQ-04 |

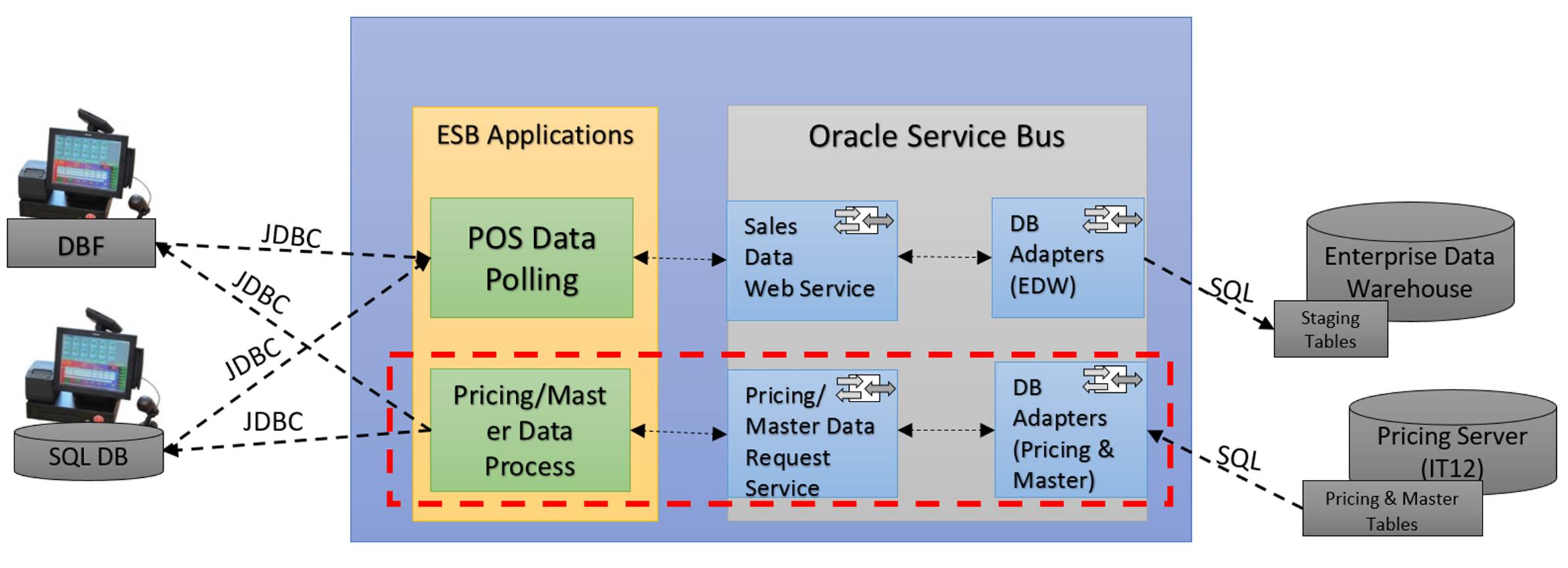
## Pricing/Master Data

### Pricing/Master Data Overall Process Flow

**Pricing/Master Data Processing**



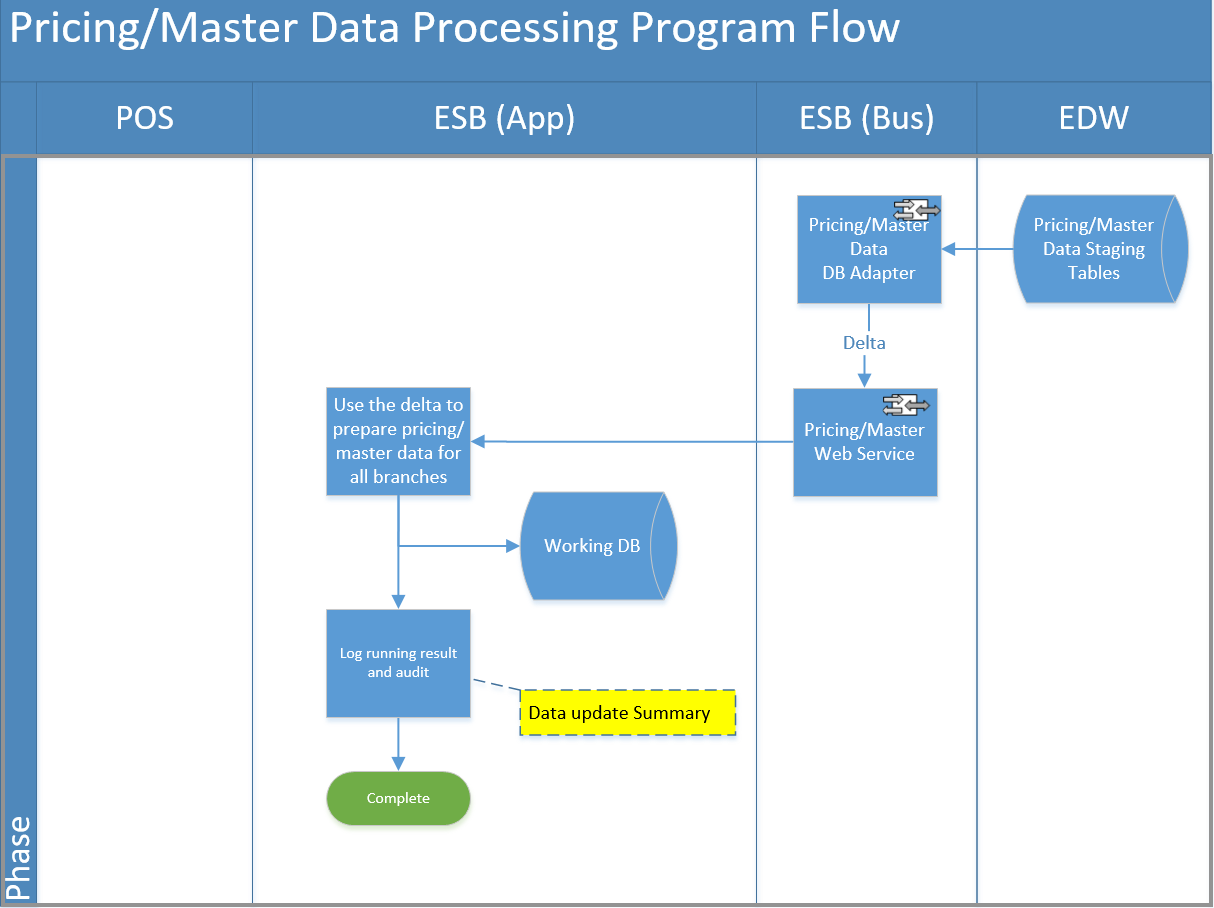
### System context



、

### Pricing data generation and download to Staging

#### Processing Flow



Task

| Activity | Description | Type | Process Group/Owner |
| --- | --- | --- | --- |
| ESB triggers the batch processing against all active POS client | 1. The pricing/master data distribution job pool is controlled by a centralized job controller which runs in a timed interval (separate the process from POS polling to avoid I/O conflicts). 2. The time trigger shall submit multiple jobs in the pool, which sequentially distribute pricing/master data to the POS client in round robin rules. 3. The job controller shall justify the job submission according to the setting from the schema control table. | Process | Sales Data Download to Staging DB – Carl |
| Ready the pricing/master data from pricing data server to service bus working database | 1. The application server will initiate an asynchronous process to periodically generate and download the pricing/master data from pricing server to service bus staging DB. The below steps would be taken.    1. Trigger existing stored procedures to prepare new pricing/master data    2. Download the prepared data to staging DB | Input | Pricing/Master Data Distribution - Wing |
| Distribution job completion | Log down a job status and process summary of the Pricing/Master data downloading processing upon staging DB. | Process | Sales Data Download to Staging DB – Carl |

Role

| Role | Description | Type |
| --- | --- | --- |
| System Administrator | The user role has the right to access the admin function of the system, e.g. job control table configuration, data source schema control configuration, parameter configuration, etc. | Admin |
| System Operator | The user role has to right to conduct job re-run, job logs and dashboard. And demander of daily summary report. | Operator |
| Pricing Server Support | The user role has to right to conduct job re-run, job logs the responsibility to receive pricing/master data related alert email. | Support |

Application

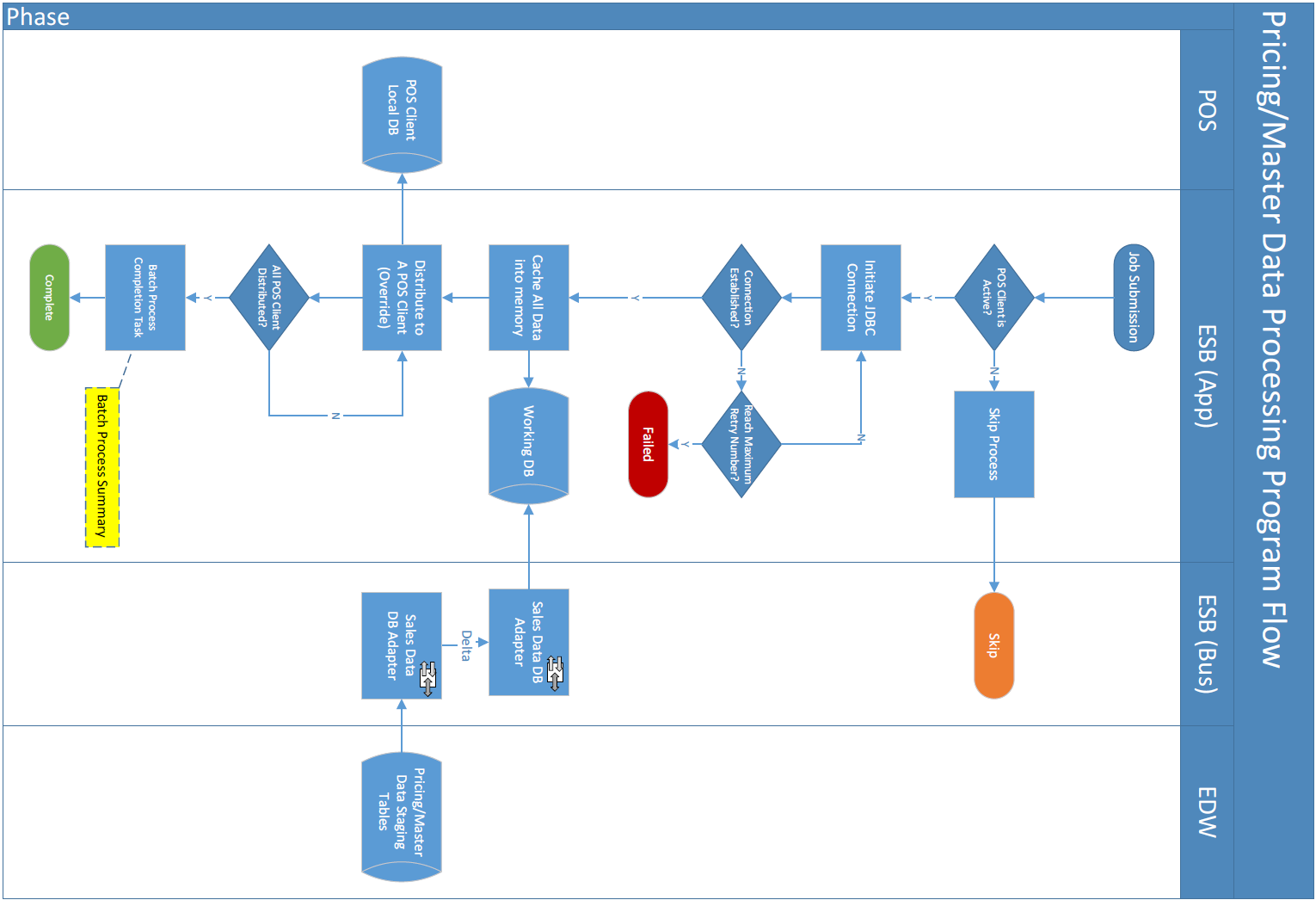
| Application | Description |
| --- | --- |
| Polling Gateway | The application module responsible of downloading data from POS client. |
| Sending Gateway | The application module to carry out data pushing to service bus interfaces. |
| ESB Interfaces | The interface for data pushing from service bus application layer staging DB to EDW. |

#### Requirement Details

| **ID** | **Name** | **Description** | **Priority** | **Owner** | **Logic** | **Parent** |
| --- | --- | --- | --- | --- | --- | --- |
| ESB-PRZ-REQ-01 | Pricing/master data processing | - | - | - | - | - |
| ESB-PRZ-REQ -01-01 | Pricing/Master Data preparation | The service bus will trigger old stored procedure to trigger the pricing/master data preparation. | Must Have | Wing | The pricing/master data preparation process will be triggered by the pre-configured time (15mins or shorter). | ESB-PRZ-REQ-01 |
| ESB-PRZ-REQ -01-02 | Pricing/Master data download to staging | After the pricing data is prepared, the service bus shall run an application download the full set pricing/master data from pricing server to staging DB. | Must Have | Wing | On triggering the stored procedure, DB adapter to monitor the change in the pricing/master data tables and simulate the same operation on staging tables. | ESB-PRZ-REQ-01 |
| ESB-PRZ-REQ -01-06 | Pricing/Master data distribution Process | Update data synchronization status in Staging database and POS client. | Must Have | Carl | After the EOD data is successfully sent to EDW, the sending result shall revert back to service bus application’s staging database, and correspondingly update the POS clients. | ESB-PRZ-REQ-01 |

### Pricing/Master data distribution (Staging - POS)

#### Process Flow



Task

| Activity | Description | Type | Process Group/Owner |
| --- | --- | --- | --- |
| ESB triggers the batch processing against all active POS client | 1. The pricing/master data distribution job pool is controlled by a centralized job controller which runs in a timed interval (separate the process from POS polling to avoid I/O conflicts). 2. The time trigger shall submit multiple jobs in the pool, which sequentially distribute pricing/master data to the POS client in round robin rules. 3. The job controller shall justify the job run rule according to the setting from the schema control table. | Process | Sales Data Download to Staging DB – Carl |
| Distributing pricing/master data to POS client | Continue on #1 task, the submitted job will distribute the corresponding pricing/master date into the target POS client. | Output | Sales Data Download to Staging DB – Carl |
| Distribution job completion | Log down a job status and process summary of the POS client for track record. | Process | Sales Data Download to Staging DB – Carl |

Role

| Role | Description | Type |
| --- | --- | --- |
| System Administrator | The user role has the right to access the admin function of the system, e.g. job control table configuration, data source schema control configuration, parameter configuration, etc. | Admin |
| System Operator | The user role has to right to conduct job re-run, job logs and dashboard. And demander of daily summary report. | Operator |
| POS Support | The user role has to right to conduct job re-run, job logs the responsibility to receive POS related alert email. | Support |

Application

| Application | Description |
| --- | --- |
| Polling Gateway | The application module responsible of downloading data from POS client. |
| Sending Gateway | The application module to carry out data pushing to service bus interfaces. |
| ESB Interfaces | The interface for data pushing from service bus application layer staging DB to EDW. |

### Requirement Details

| **ID** | **Name** | **Description** | **Priority** | **Owner** | **Logic** | **Parent** |
| --- | --- | --- | --- | --- | --- | --- |
| ESB-MST-REQ-02 | Pricing/master data processing | - | - | - | - | - |
| ESB-MST-REQ -02-01 | Pricing/Master data distribution Process | The distribution process will be submitted by the job controller, and initiate a JDBC connection to update data into POS client. | Must Have | Carl | Assume the pricing/master data has the primary key, the processing will update/merge the data by the reference key to avoid override. | ESB-MST-REQ-02 |
| ESB-MST-REQ -02-02 | Pricing/Master data distribution Process | The process checks the total number of records to ensure the distribution is successful. | Must Have | Carl | Because there’s no reference of the data indicating it’s being distributed to one POS client, the process should check against the total number of records to ensure, the processing data are pushed to POS client. | ESB-MST-REQ-02 |
| ESB-MST-REQ -02-03 | Pricing/Master data distribution Process | Data process will use incremental update towards POS clients. | Must Have | Carl | System shall compare the data in staging DB with the data in POS client, continually merge the change into POS clients. | ESB-MST-REQ-02 |
| ESB-MST -REQ -02-04 | Pricing/Master data distribution Process | Update data synchronization status in Staging database and POS client. | Must Have | Carl | After the Pricing/Master data is successfully input into POS client, the sending result shall revert back to service bus application’s staging database. | ESB-MST-REQ-02 |
| ESB-MST -REQ -02-05 | EOD process job summary | Job running summary will allow user to trace back the status and result of the schedule job. | Must Have | Carl | Update the job log to reflect the summary of the whole distribution process including   * 1. Job run status   2. Number of records processed   3. Job name, e.g. Master – Staging – POS   4. Job ID   5. Job run error stage (if any)   6. Error Category, e.g. DB, Network and Other, etc.   7. Job start/end time (regarded as Coping to POS start/end time) | ESB-MST-REQ-02 |

## Non-functional Requirement

### Requirement Details

| **ID** | **Name** | **Description** | **Priority** | **Owner** | **Logic** | **Parent** |
| --- | --- | --- | --- | --- | --- | --- |
| ESB-NF-REQ-01 | Non-functional Requirements | - | - | - | - | - |
| ESB-NF-REQ-01-01 | System dashboard view  – real time | The system dash board for real time sales data processing, shall show the summary of every full batch of jobs covering all the POS clients, and contain a brief summary of it. | Must Have | Maxim IT | Initial design should contain information below for the dashboard purpose   1. Because there are over 400 clients, so a drop box will allow user to select one bath, and system loads the summary info of the batch in the dashboard and a table 2. The table will contain below column to show the job infos    1. Data source name (branch name)    2. Job start/end time    3. Job status (Success/Failure)    4. No of records processed | ESB-NF-REQ-01 |
| ESB-NF-REQ-01-02 | POS data source control | The POS data source will have control indicators which enables the jobs easily manipulate them | Must Have | Maxim IT | 1. IsActive Indicator – the data source is active or not, if not active, the job will skip 2. Polling Start/End time – this is to avoid useless jobs occupying system resource, to limit the jobs for certain data source only be submitted during a timed range. If user’s requirement is 24x7, the initially set to 0:00-24:00 | ESB-NF-REQ-01 |
| ESB-NF-REQ-01-03 | POS data source control | For privacy concern, the password preserved in the data source control table shall be encrypted by private key | Nice to have | Maxim IT | Apply private key encryption to data source related password. | ESB-NF-REQ-01 |
| ESB-NF-REQ-01-04 | System Exception Report – Jobs Process | The job process exception shall be collected and viewable by the user according to | Must Have | Maxim IT | The job process exception report should contain the information of exception occur in process logic, e.g.   * Network error (connection lost) * DB error (cannot access) * File error (file cannot read) * Interface error (fail to invoke service) * …   These exceptions could be categorized into message severity and, the user could configure to justify above which severity, the job exception alert mail will be sent to application support team.  For example, if the “send alert email severity” = 10, if error with severity > | ESB-NF-REQ-01 |
| ESB-NF-REQ-01-05 | System Exception Report – Data Process | Before EOD process, if there’s still discrepant data accumulated in the exceptional data table, the system shall send out a data exception report to application support team. | Must Have | Maxim IT | The scenarios of data exception:   1. Data not match 2. The data cannot complete an entity relationship (e.g. order, order extra, order pay) 3. The data does not have enough information for the interface (lack of mandatory fields for the web service call) | ESB-NF-REQ-01 |
| ESB-NF-REQ-01-06 | System Exception Handling – Jobs Process | Job processing retry mechanism   * Real time processing | Must Have | Maxim IT | In real time process there won’t be any retry mechanism for MITPOS because the batch of jobs has been running in a short time interval and next run will cover the data which was not processed in the last job | ESB-NF-REQ-01 |
| ESB-NF-REQ-01-07 | System Exception Handling – Data Process | Job processing retry mechanism, will be introduce into several stage   * EOD processing | Nice to have | Maxim IT | EOD process will allow user to trigger re-run (UI based), when encounter job failure, the system admin will manual submit the jobs to process the left data in previous failure. | ESB-NF-REQ-01 |
| ESB-NF-REQ-01-08 | System Job Control | Polling schema maintenance will provide a UI for system admin to maintain the polling schema | Low priority | Maxim IT | An user interface to allow system admin to manipulate the polling schema rules including   * Polling tables * Polling columns * Target tables * Target columns * Polling direction (inbound/outbound) * … | ESB-NF-REQ-01 |
| ESB-NF-REQ-01-09 | System Job Control | The data source data could be initialized into the system by migration from Maxim existing system called “Branch Smart” | Nice to have | Maxim IT | The service bus control of the POS client branch information will be initiated from “branch smart” or synchronized data from it as single info source. | ESB-NF-REQ-01 |
| ESB-NF-REQ-01-10 | System configuration | Define system’s configuration items | Must Have | Maxim IT | Critical system parameters will be store in DB, e.g.   1. Staging DB data source user/password 2. Number of retry limit 3. Housekeeping 4. Job controller time interval for    1. Polling process controller    2. Distribution process controller    3. Pricing/master data preparation process    4. Interface web service url    5. Encryption keys    6. Support teams email | ESB-NF-REQ-01 |

### System Interface

| # | Name | Description | Priority | Owner | Business Logic | Parent |
| --- | --- | --- | --- | --- | --- | --- |
| ESB-NF-REQ-02 | System Interface | - | - | - | - | - |
| ESB-NF-REQ-02-01 | Sales Data Web Service | Web Service Interface allow external application calls to post a complete sales order data to EDO. | Must Have | Maxim IT | Data completeness - the order has a payment record for the composited key. | ESB-NF-REQ-02 |
| ESB-NF-REQ-02-02 | Pricing/Master data DB adapter | DB adapter to trigger the stored procedures to prepare pricing/master data. | Must Have | Maxim IT | Pricing/master data are generated in different groups. | ESB-NF-REQ-02 |
| ESB-NF-REQ-02-03 | Sales data DB adapters | DB adapters to query the total number of records being transferred in current EOD/real-time batch against the history tables. | Must Have | Maxim IT | Query parameter is the current batch ID (if EDW tables cannot change, use business dates) and the branch number, and sum the transferred records to compare. | ESB-NF-REQ-02 |
| ESB-NF-REQ-02-04 | Sales data DB adapters | DB adapters to query the total amount of records being transferred in current EOD/real-time batch against the history tables. | Must Have | Maxim IT | Query parameter is the current batch ID (if EDW tables cannot change, use business dates) and the branch number, and sum the total amount of transferred records in the target tables to compare. | ESB-NF-REQ-02 |
| ESB-NF-REQ-02-05 | Pricing/Master data DB adapter | DB adapter to update/insert convert log into EDW. | Must Have | Maxim IT | After the EOD data has been transferred to EDW after total record count and total amount have been validated, update the convert log in EDW to indicator the EOD data is ready for conversion. | ESB-NF-REQ-02 |

# Sign Off

|  |  |  |
| --- | --- | --- |
| Description of Deliverable: Requirement Definition Document  The requirements specification for the application. | | |
| **POS Part Sign-Off (Section: 4.2.4, 4.2.6, 4.3.5)** | | |
| Name (Print or Type) | Date | Signature |
| **CARL CHOW** |  |  |
|  | | |
| **Staging to EDW Part Sign-Off (4.2.5, 4.2.7, 4.3.6)** | | |
| Name (Print or Type) | Date | Signature |
| **POLLY KAM** |  |  |
|  | | |
| **Others Sections Sign-Off (Others)** | | |
| Name (Print or Type) | Date | Signature |
| **WING CHOI** |  |  |
|  | | |

# Appendix A – Existing Polling Servers Overview



# Appendix B – Data Process Flowchart

