Prepared by: Buzz IT Company Limited

Author: Steven Chen

Date: 20/12/2016

Version: 1.3

Maxim’s POS Polling Enterprise Service Bus Implementation Service

**EL-FY16-902**

**Sales/Master/Pricing Data Exchange**

Requirements Definition Document

# Document Control

## Document History

| Version | Date | Author | Revision Remark |
| --- | --- | --- | --- |
| 1.0 | 02/12/2016 | Steven Chen | 1st draft |
| 1.1 | 17/12/2016 | Steven Chen | Revised based on Carl, Wing and Polly’s comments. |
| 1.2 | 19/12/2016 | Steven Chen | Revised based on Carl’s comments. |
| 1.3 | 20/12/2016 | Steven Chen | Revised based on Wing’s and Polly’s comments. |

## Document/Design Owner

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

## Key Comments

| Name/Title | Choi Ka Wing | |
| --- | --- | --- |
| # | Comments |
| Comment | 1 | For every business date data, the EOD data has the last 15 minutes or more of data compared to total real-time data set. It means real time sales data process cannot process the last 15 minutes data because the POS clients EOD process will clear the POS client’s real time data staging table. |
| Response | 1 | Noted but this is the existing behavior of polling server. Service bus will initially maintain existing handling logic. |
| Comment | 2 | For current polling logic, the EOD process should conduct check sum on below 2 points only:   * Total record count * Sum amount of one target column |
| Response | 2 | Revise the detailed integrity check requirements to adjust this clarification. |
| Comment | 3 | When EOD DBF data file does not have any records, it treats as an exception and alerts the support team. |
| Response | 3 | Revised in relevant session. |
| Comment | 4 | External data source (e.g. ‘CSV’) have POS client to Staging (EOD data) and Staging to POS Client (Master data) |
| Response | 4 | Add “CSV files (2 types of csv files) accepted. File format information to be provided by Maxim’s IT.” on 4.2.3.1 task description.  Add “Pricing/Master data will distribute to POS client in CSV format. (2 types of csv file). File format information to be provided by Maxim’s IT.” On 4.3.4.1 task description. |

| Name/Title | Polly Kam | |
| --- | --- | --- |
| # | Comments |
| Comment | 1 | Missing report template format into the requirement. |
| Response | 1 | Added in appendix. |
| Comment | 2 | Unify the glossary and terms of the whole document. E.g. Staging DB with Service Bus DB; transaction with commitment control. |
| Response | 2 | Revised related terms to avoid misunderstanding. |
| Comment | 3 | Refer to the response. |
| Response | 3 | ESB-SAL-02-01 removed “double discount”.  Replace table name “order extra” to “trans” in the RDD when referring to three main polling tables.  Add statement “\*Figures appear in the reports shown in this section can be exported.” in Appendix F – Report Templates  Appendix E – EDW tables list added back “HIST\_TRANS\_MODIFIER”, ”COUPON\_SALES”, “TRANS\_MODIFIER’. Update TBC items of ITEMSTOCK to “real time”. |

| Name/Title | Carl | |
| --- | --- | --- |
| # | Comments |
| Comment | 1 | Need to add a new data source of CSV file in polling (POS client to Staging) |
| Response | 1 | Applied this requirement but may provide impact to project schedule |
| Comment | 2 | There are 2 types of DBF files, Infrasys & Pointsoft |
| Response | 2 | Noted and request Maxim’s IT to provide the format of both and confirm their format/content aligned. |
| Comment | 3 | Refer to the response |
| Response | 3 | 3.1.3.1 #1 no additional component installations on the POS client  3.1.3.1 #4 revised “covert log” to “hist\_possystem” according to Wing’s previous email  3.1.3.1 #5 Non-ESB enabled systems, e.g. “Cake order”, “Event order”  3.1.3.1 #6 complete sales order = data which has valid records in orders, trans and order pay  3.1.3.1 #9 Revised upon last meeting according to Wing’s comment  3.1.3.2 #1 Revised upon last meeting according to Wing’s comment  3.1.3.2 #3 Revised upon last meeting according to Wing’s comment  4.1 #Sales Data EOD, EOD data should check against “total record count & total amount of target column” as check sum  4.1 #Pricing/Master data, transaction = commitment control  4.2.3.1 #Task, Not “each job” the round robin rule applies on the job pool (thread pool)  4.2.3.1 #Role POS Support – re-run jobs ( from POS client to Staging) |
| Comment | 4 | Per Carl’s comment from Email received on 19/12/2016 |
| Response | 4 | 4.2.4.2 #ESB-SAL-02-01, Correct key combination for hist-paysum & his\_payfig to be provided by Maxim’s.  4.2.4.2 #ESB-SAL-02-01, has agreed with Polly, the check sum of both EOD and real time sales data focuses on total record count and total amount.  4.2.5.2 #ESB-SAL-REQ -03-01, agreed and convert log was the previous requirement typo, has already changed to hist\_possystem. If there’s other rules, please provide the details  4.2.6.2 #ESB-SAL-REQ -04-01 Add back check sum into session 4.2.5.2 to reflect carl’s comment in ESB-SAL-REQ -03-04  44.2.6.2 #ESB-SAL-REQ -04-04, this part is revised. Just Carl’s version is not up-to-date.  4.3.3.2 #ESB-PRZ-REQ -01-02 Please provide details to complete the EOD tables descriptions in the appendix reflecting which tables are full set and which are delta.  4.3.5 #ESB-MST-REQ -02-02 Noted. We will use application’s transaction control to intimate current commitment control in the stored procedures. |

# Table of Content

[Document Control 2](#_Toc469739587)

[Document History 2](#_Toc469739588)

[Document/Design Owner 2](#_Toc469739589)

[Key Comments 2](#_Toc469739590)

[Table of Content 5](#_Toc469739591)

[1 Background 8](#_Toc469739592)

[1.1 Document Purpose 8](#_Toc469739593)

[1.2 Document Scope 8](#_Toc469739594)

[1.3 Document Audience 8](#_Toc469739595)

[1.4 Terms & Abbreviations 9](#_Toc469739596)

[1.5 Reference Materials 9](#_Toc469739597)

[2 Executive Summary 10](#_Toc469739598)

[3 Business Context & Concept 11](#_Toc469739599)

[3.1 Business Rationale, Objectives & Considerations 11](#_Toc469739600)

[3.1.1 Objectives 11](#_Toc469739601)

[3.1.2 Expected Business Benefits 11](#_Toc469739602)

[3.1.3 Considerations 12](#_Toc469739603)

[3.1.3.1 Assumptions 12](#_Toc469739604)

[3.1.3.2 Constraints 13](#_Toc469739605)

[3.1.3.3 Dependencies 14](#_Toc469739606)

[3.1.3.4 Issues 14](#_Toc469739607)

[3.1.3.5 Risks 14](#_Toc469739608)

[3.2 Value Propositions 15](#_Toc469739609)

[4 Business Process 16](#_Toc469739610)

[4.1 Business Process Scope 16](#_Toc469739611)

[4.2 Sales Data 18](#_Toc469739612)

[4.2.1 Sales Data Overall Process Flow 18](#_Toc469739613)

[4.2.2 System context 20](#_Toc469739614)

[4.2.3 Sales Data Real Time Polling (POS – Staging) 21](#_Toc469739615)

[4.2.3.1 Process Flow 21](#_Toc469739616)

[4.2.3.2 Functional Requirement Details 24](#_Toc469739617)

[4.2.4 Sales Data Real Time Polling Flow (Staging - EDW) 27](#_Toc469739618)

[4.2.4.1 Processing Flow 27](#_Toc469739619)

[4.2.4.2 Functional Requirement Details 30](#_Toc469739620)

[4.2.5 Sales Data End-of-day (POS – Staging) 34](#_Toc469739621)

[4.2.5.1 Process Flow 34](#_Toc469739622)

[4.2.5.2 Functional Requirement Details 38](#_Toc469739623)

[4.2.6 Sales Data End-of-day (Staging – EDW) 41](#_Toc469739624)

[4.2.6.1 Process Flow 41](#_Toc469739625)

[4.2.6.2 Functional Requirement Details 44](#_Toc469739626)

[4.3 Pricing/Master Data 48](#_Toc469739627)

[4.3.1 Pricing/Master Data Overall Process Flow 48](#_Toc469739628)

[4.3.2 System context 49](#_Toc469739629)

[4.3.3 Pricing data generation and download to Staging 50](#_Toc469739630)

[4.3.3.1 Processing Flow 50](#_Toc469739631)

[4.3.3.2 Requirement Details 53](#_Toc469739632)

[4.3.4 Pricing/Master data distribution (Staging - POS) 54](#_Toc469739633)

[4.3.4.1 Process Flow 54](#_Toc469739634)

[4.3.5 Requirement Details 57](#_Toc469739635)

[4.4 Non-functional Requirements 59](#_Toc469739636)

[4.4.1 Requirement Details 59](#_Toc469739637)

[4.4.2 System Interface 65](#_Toc469739638)

[5 Sign Off 67](#_Toc469739639)

[6 Appendix A – Existing Polling Servers Overview 68](#_Toc469739640)

[7 Appendix B – Data Process Flowchart 68](#_Toc469739641)

[8 Appendix C – EDW Virtual Branch Sales Inbound Tables 68](#_Toc469739642)

[9 Appendix D – POS Client Polling Table 68](#_Toc469739643)

[10 Appendix E – EDW Tables 70](#_Toc469739644)

[11 Appendix F - Report Templates 71](#_Toc469739645)

# 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 detailed design documents and the implemented solution.

The Requirements Definition Document will provide a common understanding of all of the requirements for all project stakeholders. It covers 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 part of the deliverables in 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, processes design/flow, system contexts (outlining key applications used by the business unit/domain and the relationships that exist among them), functional requirements, use cases (describing the functional behavior of the impacted applications) and non-functional requirements (describing the non-functional behavior of the supporting operations). The technical designs and specifications of the impacted applications are not included in this document.

## Document Audience

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

## Terms & Abbreviations

|  |  |
| --- | --- |
| **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 |
| Staging DB | The service bus database to stage the polling data |

## Reference Materials

| Document Names |
| --- |
| Maxim’s 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 a new platform using Oracle Enterprise Services Bus technology. In the new ESB polling system (ESB system), three data processing flows will be implemented:

* Sales data real time processing to EDW
* Sales data EOD processing to EDW
* Synchronize master data to POS clients (e.g. Pricing/Master)

The ESB system will use JDBC to connect to the databases of existing POS clients in outlets by pre-configured connection information in order to collect sales data from the existing POS systems and update the pricing/master data back to the existing POS clients. Referring to the polling logic found in the production IT51 server, for any connection error, the ESB system will log down the error and retry data synchronization. After reaching maximum retry count, the ESB system will halt the synchronization for that particular POS client and generate alert to related parties for follow up.

All POS clients’ connection settings are configurable and maintainable by Maxim’s IT. It makes the adding of new POS client easy and no alteration of programming code is required. The ESB system can invoke several configurable concurrent threads to poll/push the data to/from POS clients concurrently for maximizing system 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. | Use Service Bus as an enterprise application layer to take over the responsibilities & functionalities of real time polling from Maxim’s POS clients which are currently done by SQL Server linked servers and SQL agent jobs. |
| 2 | Bridge the POS Clients and EDW to Transfer EOD Sales Data through Service Bus. | Use Service Bus as an enterprise application layer to take over the responsibilities & functionalities of EOD data polling from Maxim’s POS clients which are currently done by SQL Server linked servers and SQL agent jobs. |
| 3 | Enable the Pricing/Master Data Distribution to POS Clients through Service Bus. | Use Service Bus as an enterprise application layer to take over the responsibilities & functionalities of Pricing/Master data pushing to Maxim’s POS clients which are currently done by SQL Server linked servers and SQL agent jobs. |

**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 | Service bus’s application layer will maintain JDBC connection to collect data from the DB of POS clients.  No additional component installation required on the POS client in this phase. | Carl |
| 2 | Sales & Sales EOD Data Pushing to EDW | Service bus will write sales and sales EOD data into corresponding staging tables in EDW. EDW will then trigger conversion process against these data to formalize them into other tables. 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 | “hist\_possystem” Record Monitoring for EOD process | “hist\_possystem” record used as an acknowledgement of EOD processing. | Carl |
| 4 | History Records | Assume the history tables are ready for download in POS client once the related “hist\_possystem” record found. | Carl |
| 5 | Exception Handling & Report | Oracle ESB Exception Report shows only the exception for those ESB enabled POS system. For those non-ESB based POS system (cake orders, event orders?), 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 | Sales data from MITPOS could be considered as completed sales order data which has valid records in orders, trans and order pay. | Carl |
| 7 | Void order data – MITPOS | Voided sales order would become another reverse order records with the void flag marked to ‘1’ from POS client to service bus. | Carl |
| 8 | Direct DB-to-DB 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 | Pricing/Master update should be maintained the grouped data together in one transaction/commitment. 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 | As records from POINTSOFT POS client will 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. Records under the key combination (branch code, order no., business date) exist in tables of orders, trans and order pay. This consider as the order completeness 2. The data transfer will check sum on total record count and total amount of the target column | 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 carry out record count check. For summing a specific amount field for comparison is not able to carry out. | Wing |
| 3 | Real Time Sales Data Polling | Real Time sales data-polling process is required to run within configurable time (Start/End time). When the system time is out of the period, it should skip the job submission.  According to Maxim’s IT, the job trigger should strictly refer to the start time but actual end time should refer to EOD process, the configured end time is for reference.  The polling process initiates DB connection in a timed interval even though there is a possibility that the POS client is offline. | Carl |
| 4 | POS Client Pricing/Master Data Update Mechanism | No need to delete record in 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 does not exist, just insert NEW records.  For DELETE action, system only logical mark delete the record but NOT physically delete the record. | Carl |
| 5 | Pricing/Master Data Update in Pricing Server | In current pricing server, it requires the triggering of stored procedures to generate pricing data. To adopt existing logic, service bus will have DB adapter implemented to trigger this process by time interval rather than using 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, sales data in POS clients will upload to 2 data consumers in real time.   1. Sales operation team – for data backup, auditing and other business process (if Sales server still require the data). 2. EDW team – use partial data for data conversion and analysis purpose.   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-Day Processing | 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 (indicating that client has finished local data processing) is ready for polling.  In the EOD processing, the service bus shall check the total record count and total amount of the target column to ensure the check sum for the POS client’s EOD processing to staging without changing. |
| Master/Pricing Data Processing | Master data and Pricing data shall be generated together in 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/commitment control 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 by 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 |

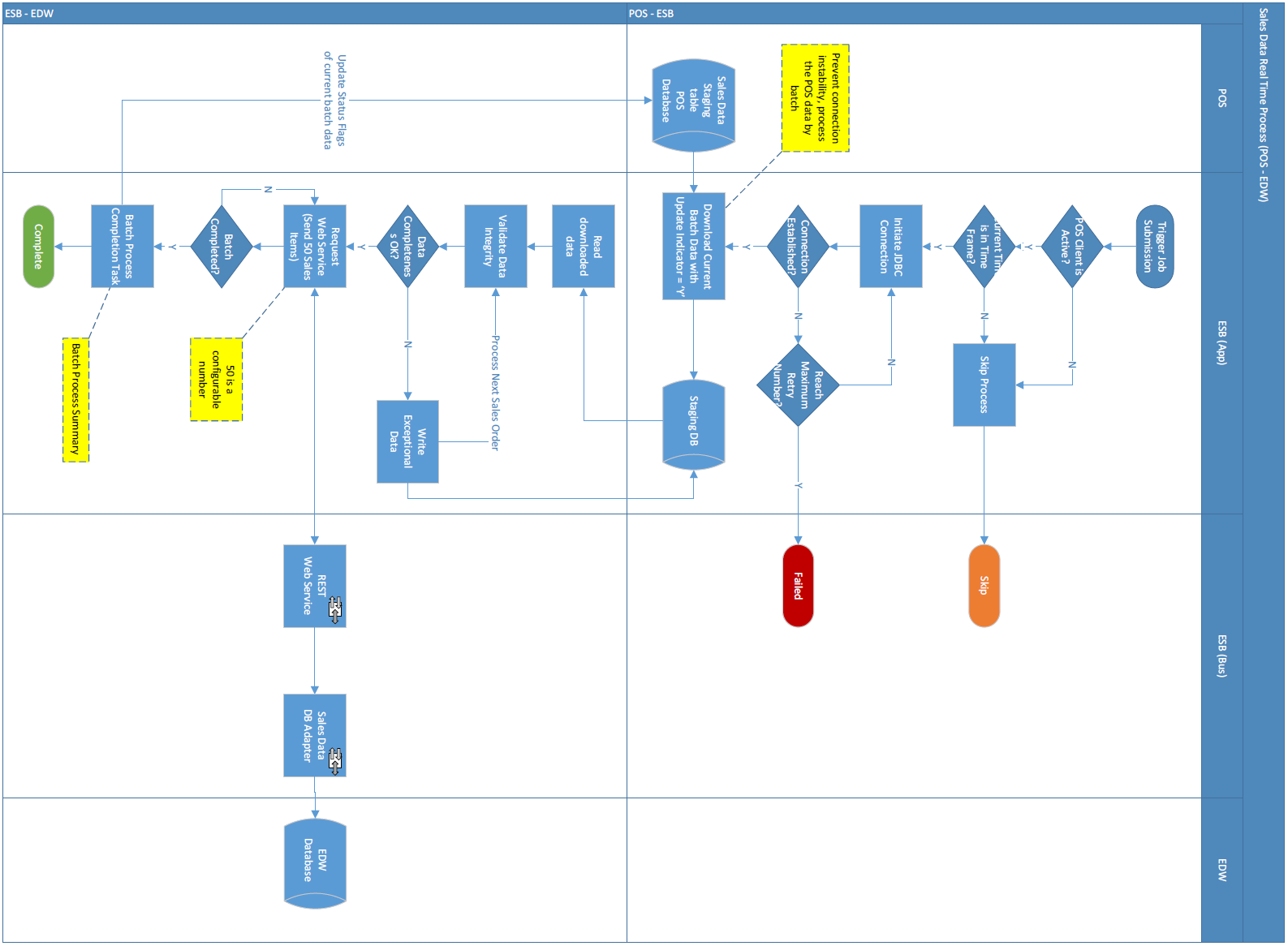
Business Process

|  |  |  |
| --- | --- | --- |
| 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’s Headquarters’ EDW in real time. |
| Sales Data End-of-day Process | For each POS client, execute once (at most) per day to copy data from POS client to EDW. | To transfer EOD data from each POS clients to EDW tables by on-demand trigger. |
| Master/pricing Data Generation and Update to POS client | Whenever there are updates on master/pricing data, push the updated data to corresponding POS client. | To copy data from pricing/master data DB and distribute them to POS client according to the pricing group setting. |

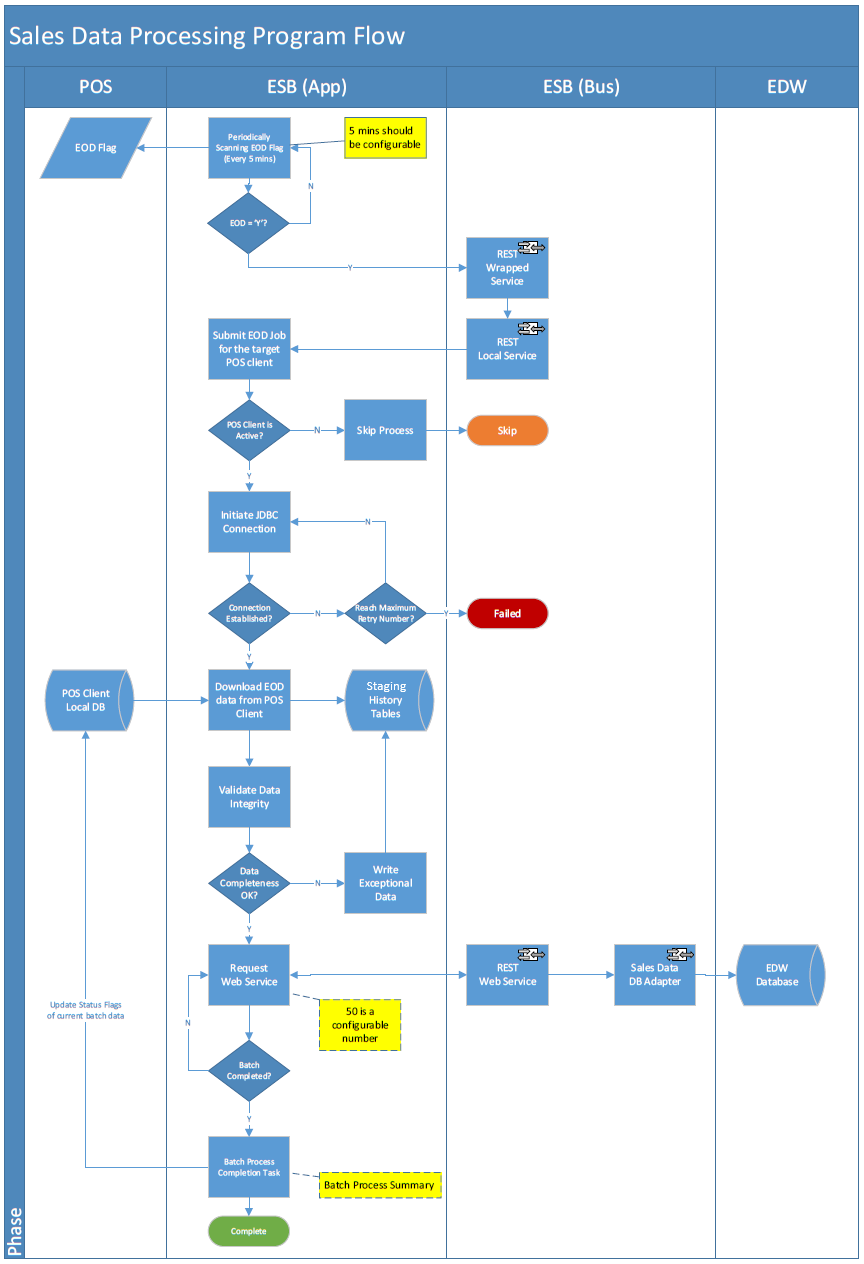
## Sales Data

### Sales Data Overall Process Flow

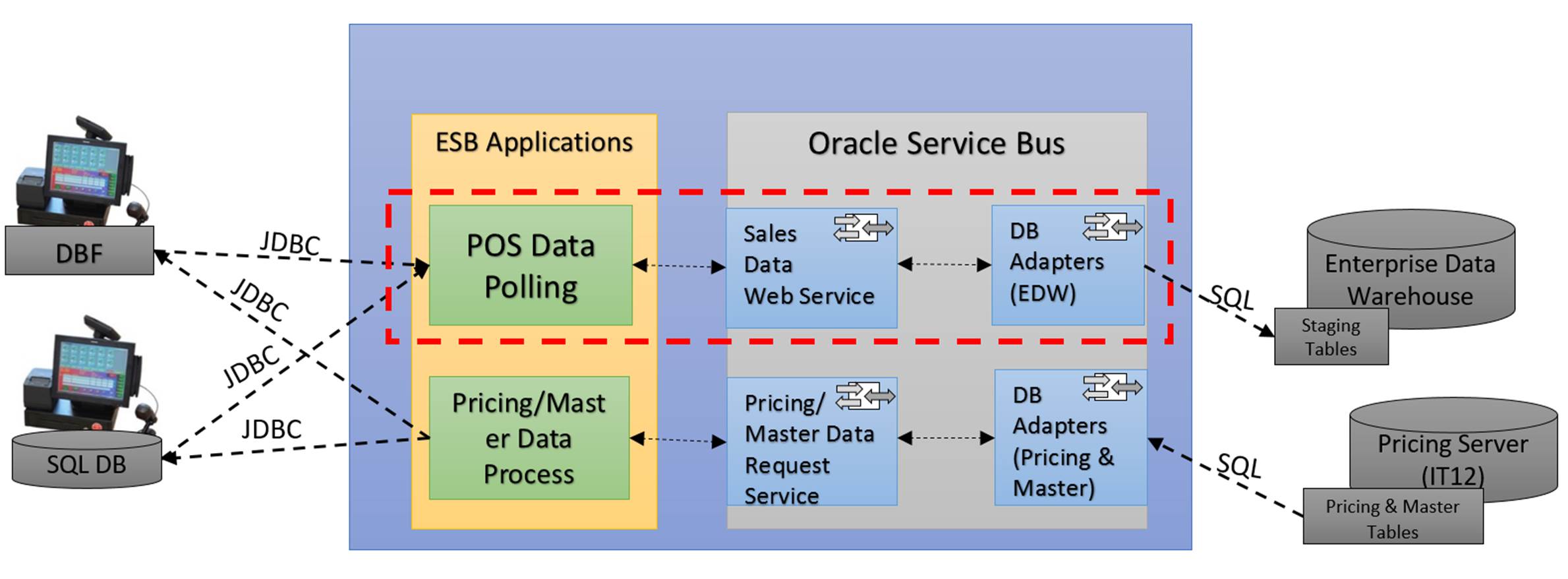
**Sales Data Real Time Processing**



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

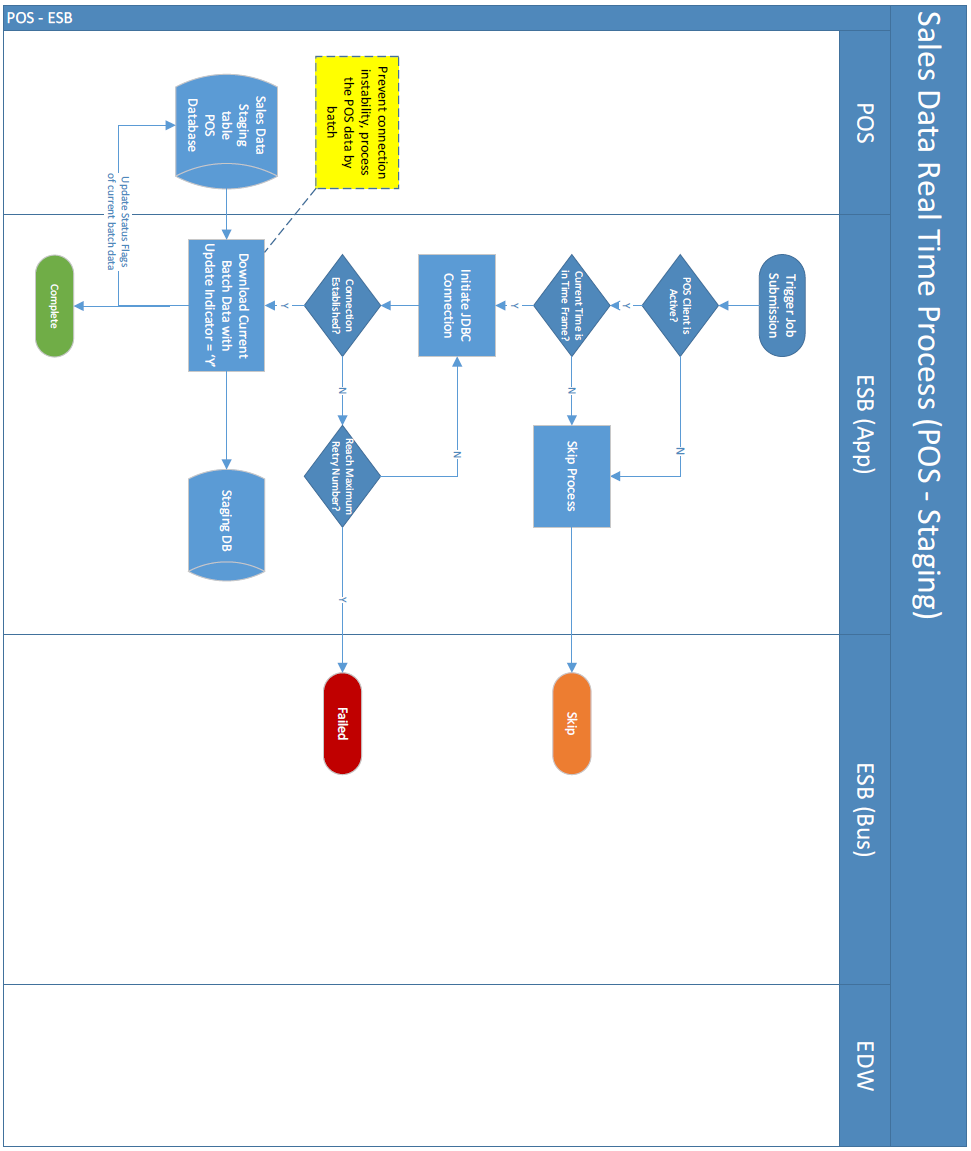


### System context



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

#### Process Flow



Task

| Activity | Description | Type | Process Group - Owner |
| --- | --- | --- | --- |
| ESB Triggers the Batch Processing Against POS Clients | 1. The POS polling job pool is controlled by a centralized job controller which runs periodically 2. The job controller job will submit multiple jobs in the pool and the set of jobs 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 polling in POS client’s DB. 2. If the data source is a DBF file, the job will scan the target directory configured in the “poll\_schema\_info” table, and use the DBF data to override the data in the staging tables. 3. CSV files (2 types of csv files) accepted. File format information to be provided by Maxim’s IT.   Note: Assumed that POINTSOFT POS client should use FTP transfer the DBF file to target network directory (FTP server). | 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 the right to conduct job re-run, job logs and dashboard; demand of daily summary report. | Operator |
| POS Support | The user role has the right to re-run jobs and query the job logs (from POS client to Staging) the responsibility to receive POS related alert email. | Support |

Application

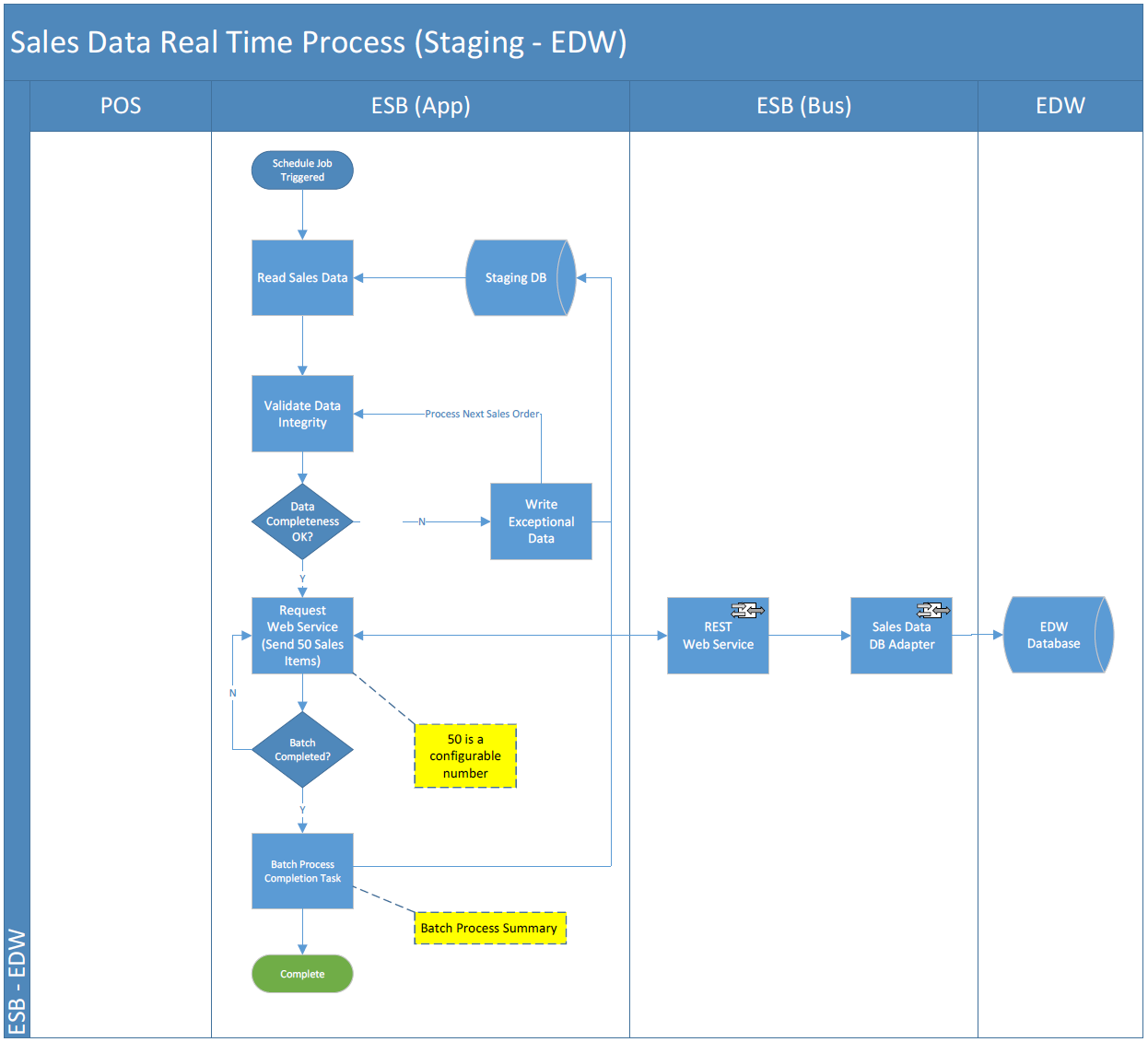
| Application | Description |
| --- | --- |
| Polling Gateway | The application module responsible of downloading data from POS client. |
| Sending Gateway | The application module to carry out data pushing to service bus interfaces. |
| ESB Interfaces | The interface for data pushing from the Staging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

#### Functional Requirement Details

| **ID** | **Name** | **Description** | **Priority** | **Owner** | **Logic** | **Parent** |
| --- | --- | --- | --- | --- | --- | --- |
| ESB-SAL-001 | Sales Data Real Time polling to Staging DB | Collect the sales data in POS clients 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 7x24 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 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 is 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 found.  For Sales Polling, only "whole" sales transactions sent to EDW for Sales data Polling. Partial sales records (e.g. POINTSOFT POS) should NOT send to EDW. The data from POINTSOFT client will stage 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 transfer success, the processing job updates the polling flag status back to POS client indicating the batch of data has already 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 | The real time sales data processing job (Staging to EDW) submitted to process the real time sales data in the Staging DB of target POS client. | Process | Send POS sales data from Staging DB to EDW – Polly |
| Validate Data Integrity | The job will continue to validate (refer to requirement details of Sales Data real time processing in below section 4.2.4.2 #ESB-SAL-02-03) the polled sales data’s integrity by order in the staging table of current batch. If the whole batch data’s completeness guaranteed, the job will move to next steps. | Process | Send POS sales data from Staging DB to EDW – Polly |
| Send Sales Data to EDW | 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.   If the transfer is successful, the response result will update into staging database, and lately reverted to Staging DB upon the job’s completion. | Output | Send POS sales data from Staging DB to EDW - Polly |
| Job Completion | 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 the right to conduct job re-run, job logs and dashboard; demand of daily summary report. | Operator |
| Service Bus Operator | The user role has the right to conduct job re-run, job logs the responsibility to receive EDW related alert email. | Support |

Application

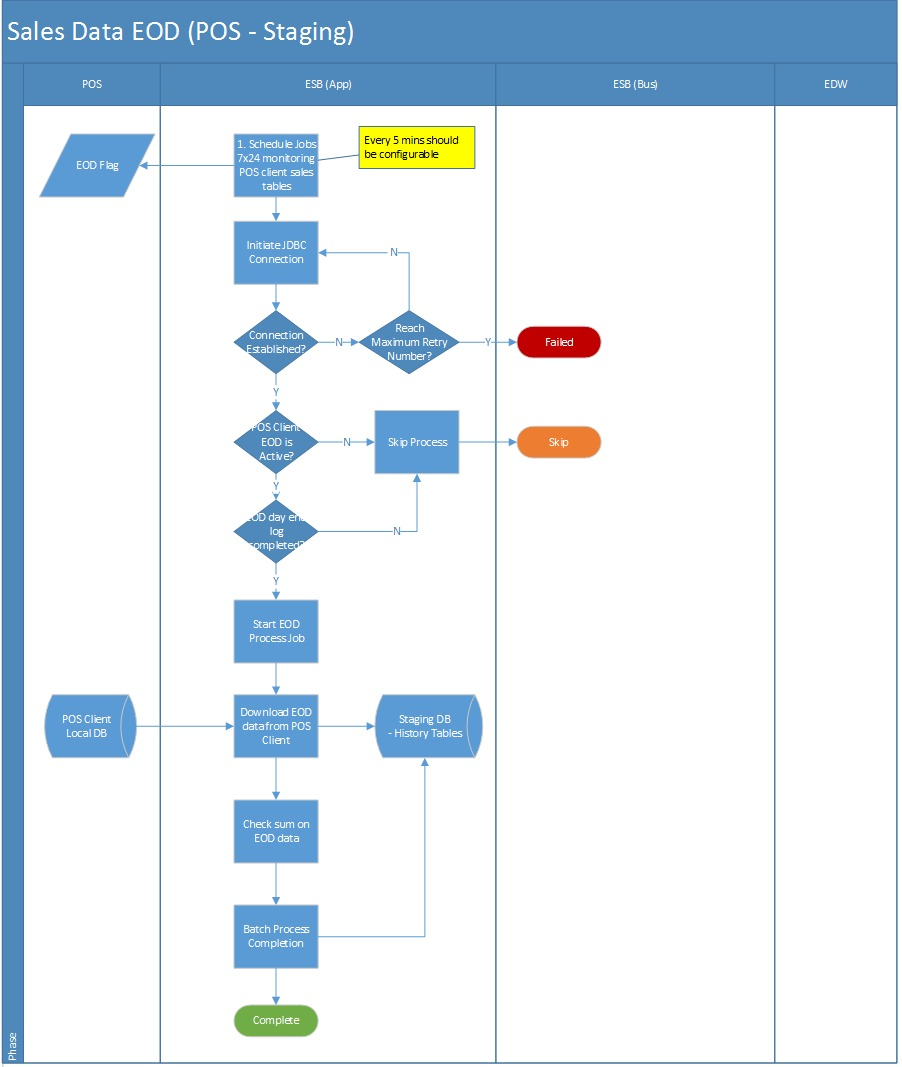
| Application | Description |
| --- | --- |
| Polling Gateway | The application module responsible of downloading data from POS client. |
| Sending Gateway | The application module to carry out data pushing to service bus interfaces. |
| ESB Interfaces | The interface for data pushing from the Staging DB of service bus application layer 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 real time sales data processing job (Staging to EDW) 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) For each sales transaction, system will treat “orders”, “trans” and “order\_pay” are mandatory where as “coupon” is optional.  b) Tables are linked by:  - Business Date  - Branch Code  - Order No.  c) All sales tables linked up by the above keys. | 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 (check\_logs) * Coupon sales (coupon\_sales) * Orders (orders) * Orders pay (orders\_pay) * Redeem coupon (redeem\_coupon) * Supp (supp) * Trans (trans) * Transaction ecard (trans\_ecard) * Transaction modifier (trans\_modifier) |  |
| ESB-SAL-02-03 | Data Processing to EDW | After verifying whether the sales data is completed. Send the completed data to EDW. | Must Have | Polly | The service bus application will use “business date, branch code, order no.” to check the integrity of data. When the data in tables of “orders”, “trans” and “order pay” (but not limit to) 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. Because the data updating logic of POINTSOFT is full table override, the service bus application will wait until the data of a sales order to complete, and send to EDW.  There will be an indicator in the staging database reflecting the records sent or not. Until the polling cut-off time (4:00am, configurable), 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, and the job will end with error data log. | Must Have | Polly | The service bus has the limitation of network; the number of records shall be transferred block by block (e.g. 50) into EDW through the service. This number shall be a configurable item in the system.  When the service response success return code, update to the staging tables indicating the corresponding records 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. | The enterprise service bus processes shall monitor the update of the “hist\_possystem” which was written in POS clients.  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 | 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 Data Integrity | After successfully written into Staging DB, the processing job will conduct a check sum upon the data set with below criteria:   1. Number of record count received 2. Total Amount check sum 3. 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 | 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 the right to conduct job re-run, job logs and dashboard; demands the receipt of daily summary report. | Operator |
| POS Support | The user role has the right to conduct job re-run, job logs the responsibility to receive POS related alert email. | Support |

Application

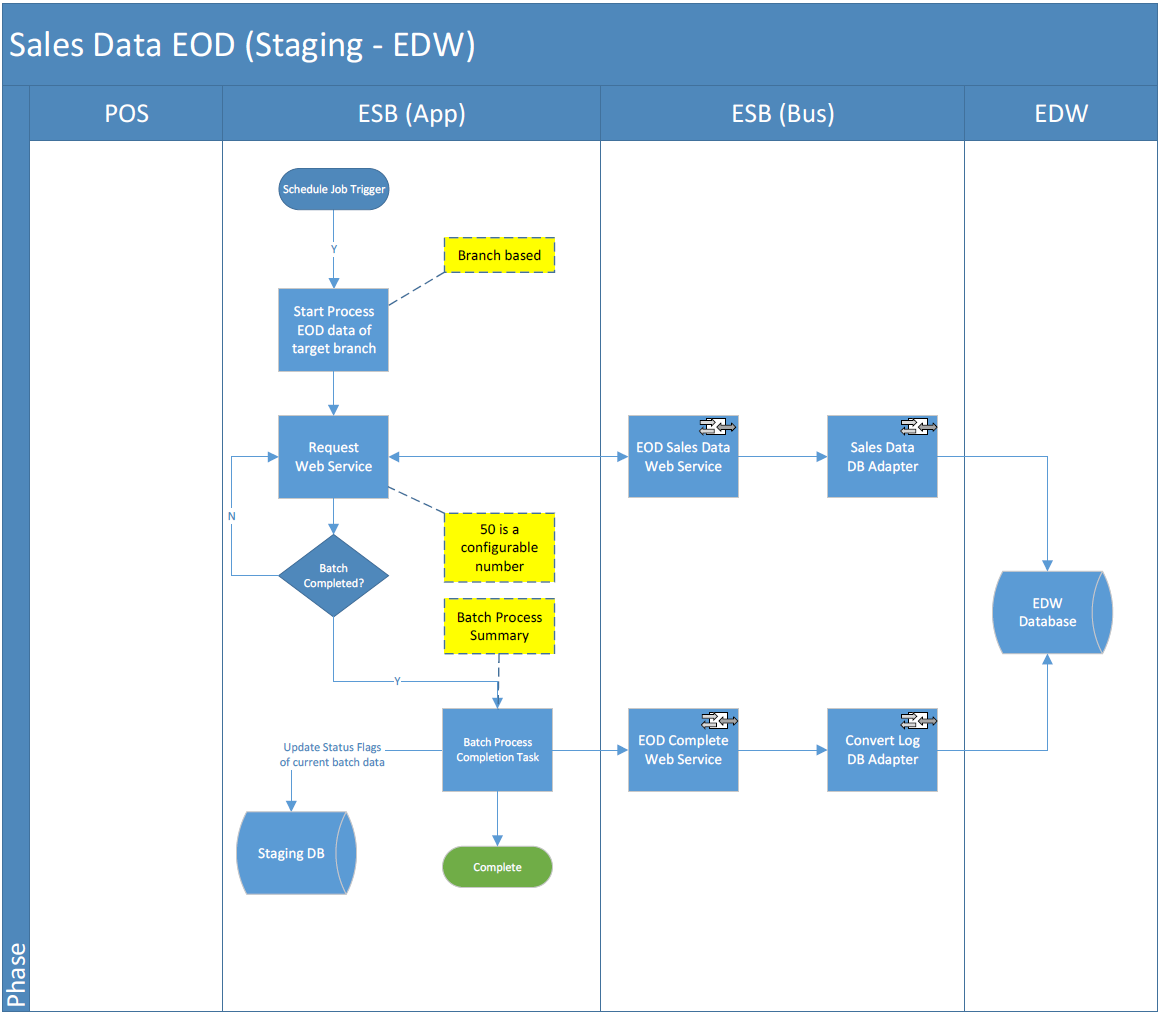
| Application | Description |
| --- | --- |
| Polling Gateway | The application module responsible of downloading data from POS client. |
| Sending Gateway | The application module to carry out data pushing to service bus interfaces. |
| ESB Interfaces | The interface for data pushing from service bus’s 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 hist\_possystem 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. Therefore, 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 successfully downloaded to Staging, synchronization result is no need to update back to POS client but logical check sum is required as below   1. Number of record count received 2. Total Amount check sum 3. 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) | 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 | 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) | 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. | Upon the completion of downloading EOD data of the target branch into staging, the EOD sales data process (Staging to EDW) triggered to read those data (same branch) from Staging DB. | Process | Send POS sales data from Staging DB to EDW - Polly |
| Send the Sales Data to EDW | 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 | If the transfer is successful, the response result updated into staging database, and lately reverted to staging database upon service call response. | Output | Send POS sales data from Staging DB to EDW - Polly |
| Validation on Sent Data | 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 completed. | Output | Send POS sales data from Staging DB to EDW - Polly |
| Job Completion | 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 the right to conduct job re-run, job logs and dashboard; demand of daily summary report. | Operator |
| Service Bus Operator | The user role has the 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. |

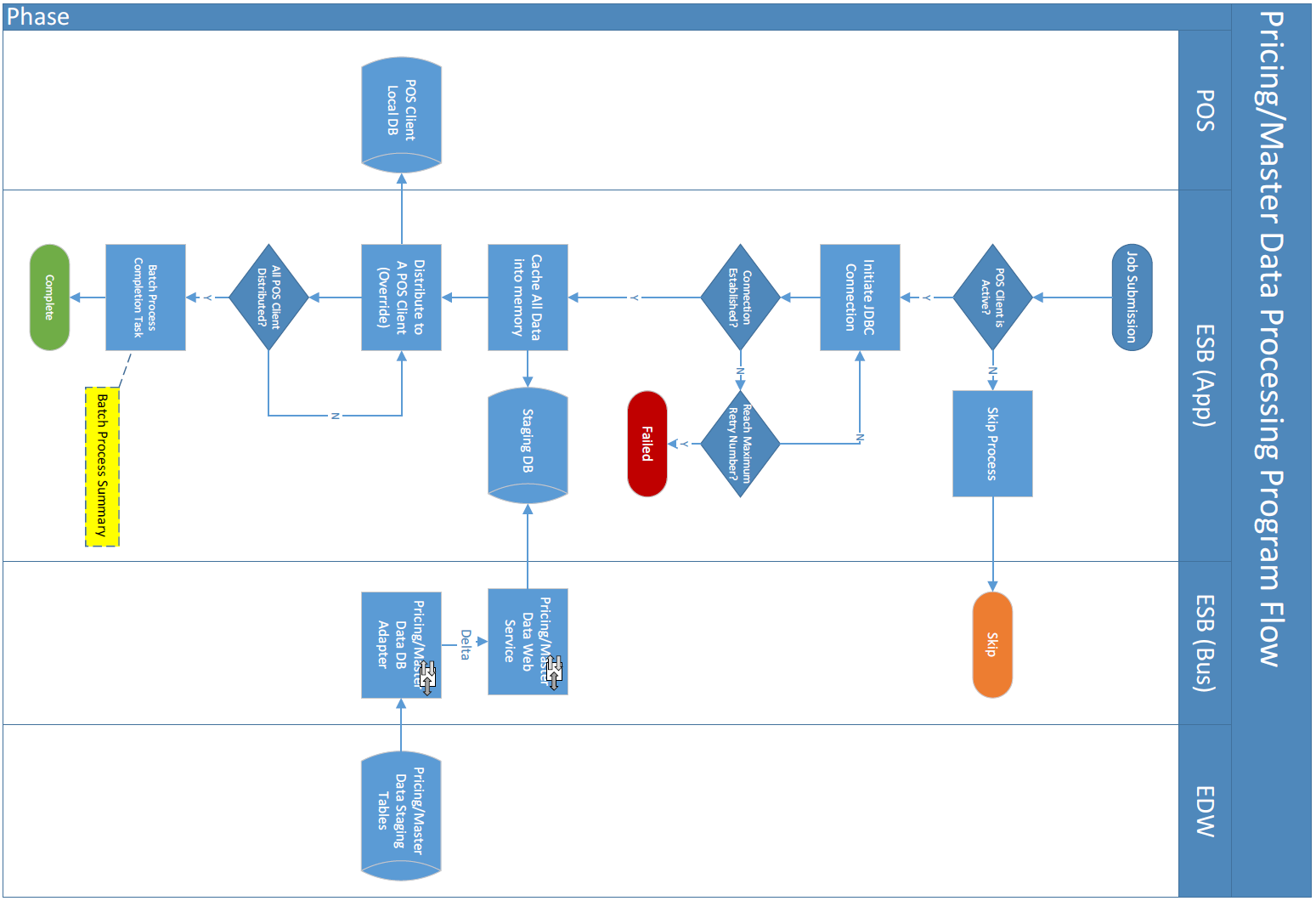
#### Functional Requirement Details

| **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 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 committed by POS client. Data will then copied to EDW directly 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 belonging 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 transaction date < last cut-off time, then business date = transaction data - 1 2. When record transaction date > last cut-off time, then business date = transaction date   For example,  EOD records in HIST\_ORDER  Business Date, transaction date  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) ---------  Case 1: 25-Nov-2016, 25-Nov-2016 02:12 🡺 Business date changes to 24-Nov-2016  Case 2: 25-Nov-2016, 25-Nov-2016 06:33 🡺 Business date changes to 25-Nov-2016  Case 3: 25-Nov-2016, 24-Nov-2016 03:33 🡺 Business date changes to 23-Nov-2016 | ESB-SAL-REQ-04 |
| ESB-SAL-REQ -04-03 | POS Client EOD Data Process | The service bus EOD processing job will validate the data with basic check sum logic upon total record count and total amount of target column | Must Have | Polly | Check sum   * Total record count * Total amount of target column   *Note: refer to udsp\_check\_upload\_data* | ESB-SAL-REQ-04 |
| ESB-SAL-REQ -04-04 | 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 retried with retry limit) 2. Data cannot be inserted 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 retried with retry limit 4. When file-based interface (DBF, CSV) contain zero record. | ESB-SAL-REQ-04 |
| ESB-SAL-REQ -04-05 | POS Client EOD Data Process | Update data synchronization status in Staging database and POS client | Must Have | Polly | After the EOD data successfully sent to EDW, the sending result shall revert to service bus application’s staging database, and correspondingly update the POS clients. | ESB-SAL-REQ-04 |
| ESB-SAL-REQ -04-06 | 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 | Update the job log to reflect the summary of the whole EOD process including:  Job run status  Number of records processed  Job name, e.g. EOD – Staging – EDW  Job ID  Job run error stage (if any)  Error Category, e.g. DB, Network and Other, etc.   1. Job start/end time (regarded as Coping to EDW start/end time) | ESB-SAL-REQ-04 |

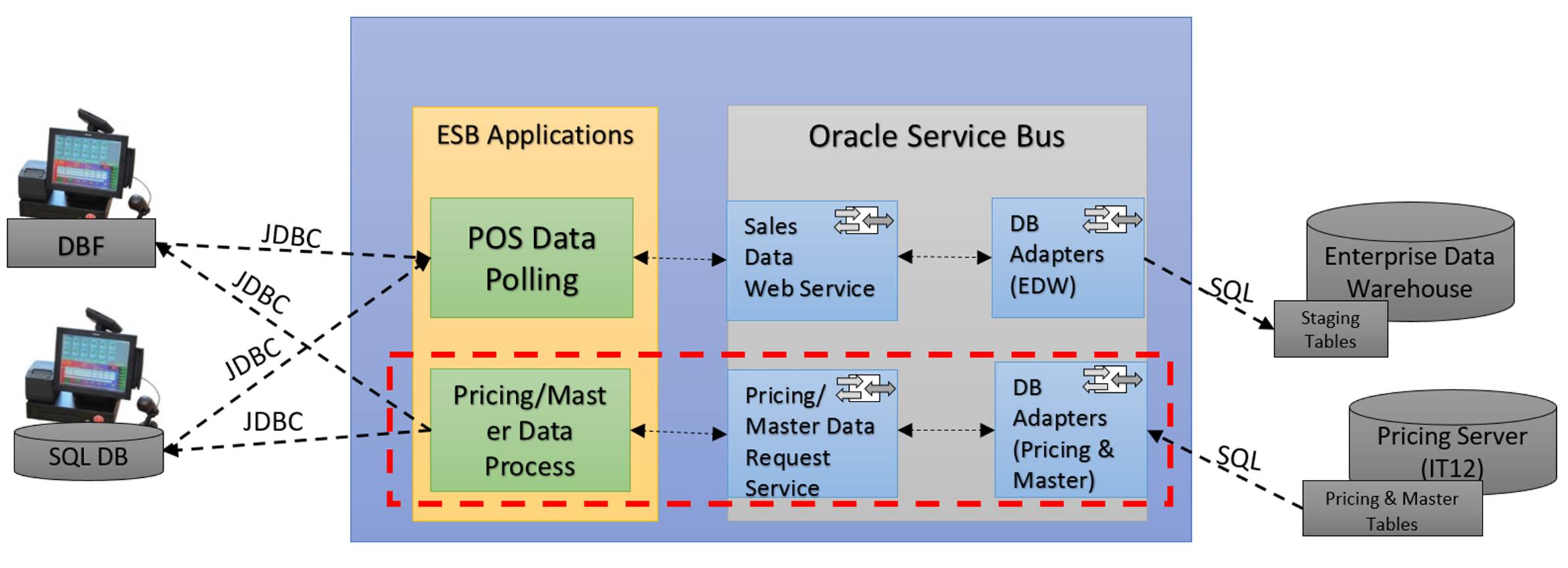
## Pricing/Master Data

### Pricing/Master Data Overall Process Flow

**Pricing/Master Data Processing**



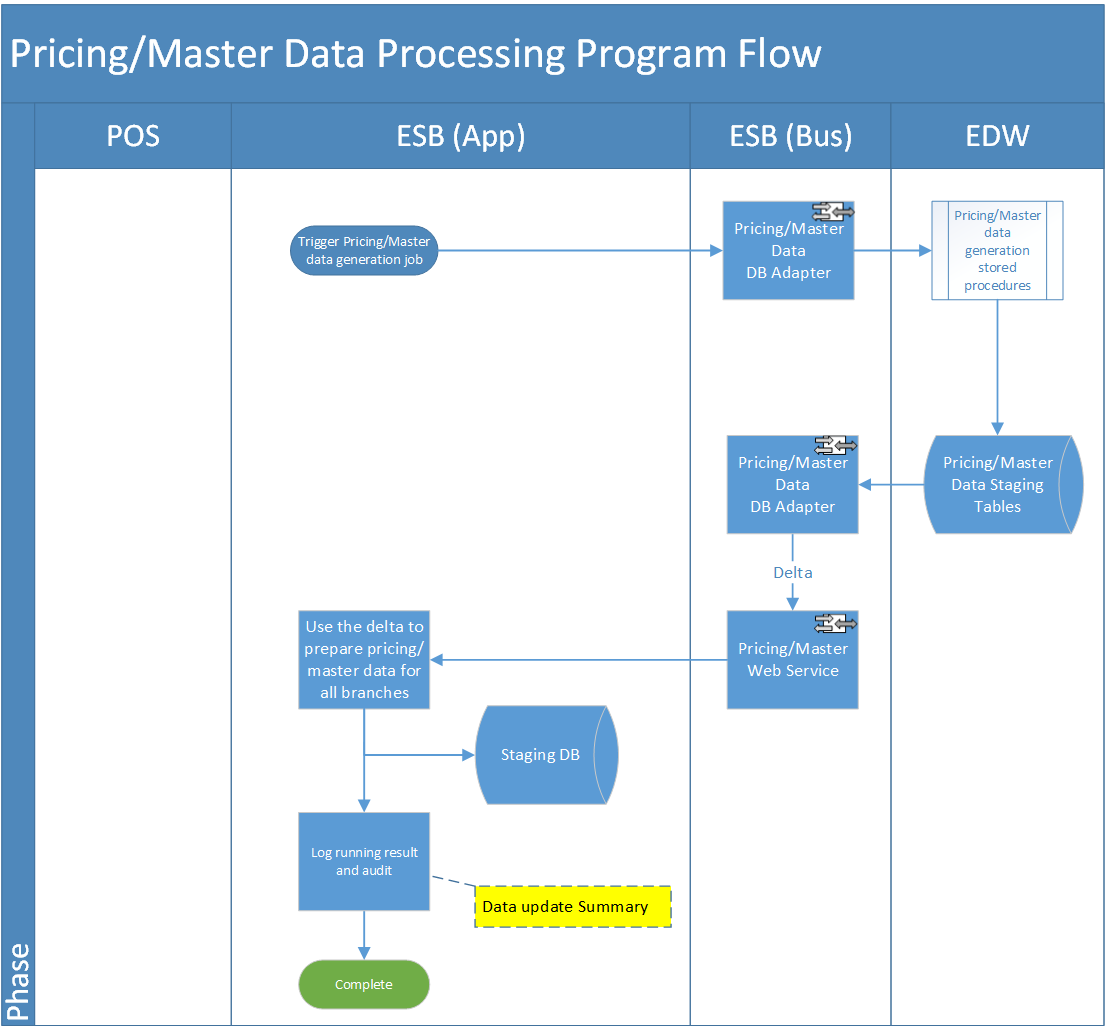
### System context



、

### 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 Clients | 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).  The job shall execute the relevant stored procedures (owned by Maxim’s Pricing/Master server) to prepare and generate pricing/master data in IT12/IT50. | Process | Sales Data Download to Staging DB – Carl |
| Ready the Pricing/Master data from Pricing data Server to Service Bus Working Database | The Service Bus DB Adapter will monitor the data delta and periodically pull the pricing/master data from pricing server to service bus staging DB. The below steps would be taken.   * 1. Monitoring delta change in target table and invoke the consumer service   2. Download the prepared data to staging DB (via SQL or Service) | Input | Pricing/Master Data Distribution - Wing |
| Service Bus DB Adaptor Monitoring | 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 the right to conduct job re-run, job logs and dashboard; demand of daily summary report. | Operator |
| Pricing Server Support | The user role has the right to conduct job re-run, job logs the responsibility to receive pricing/master data related alert email. | Support |

Application

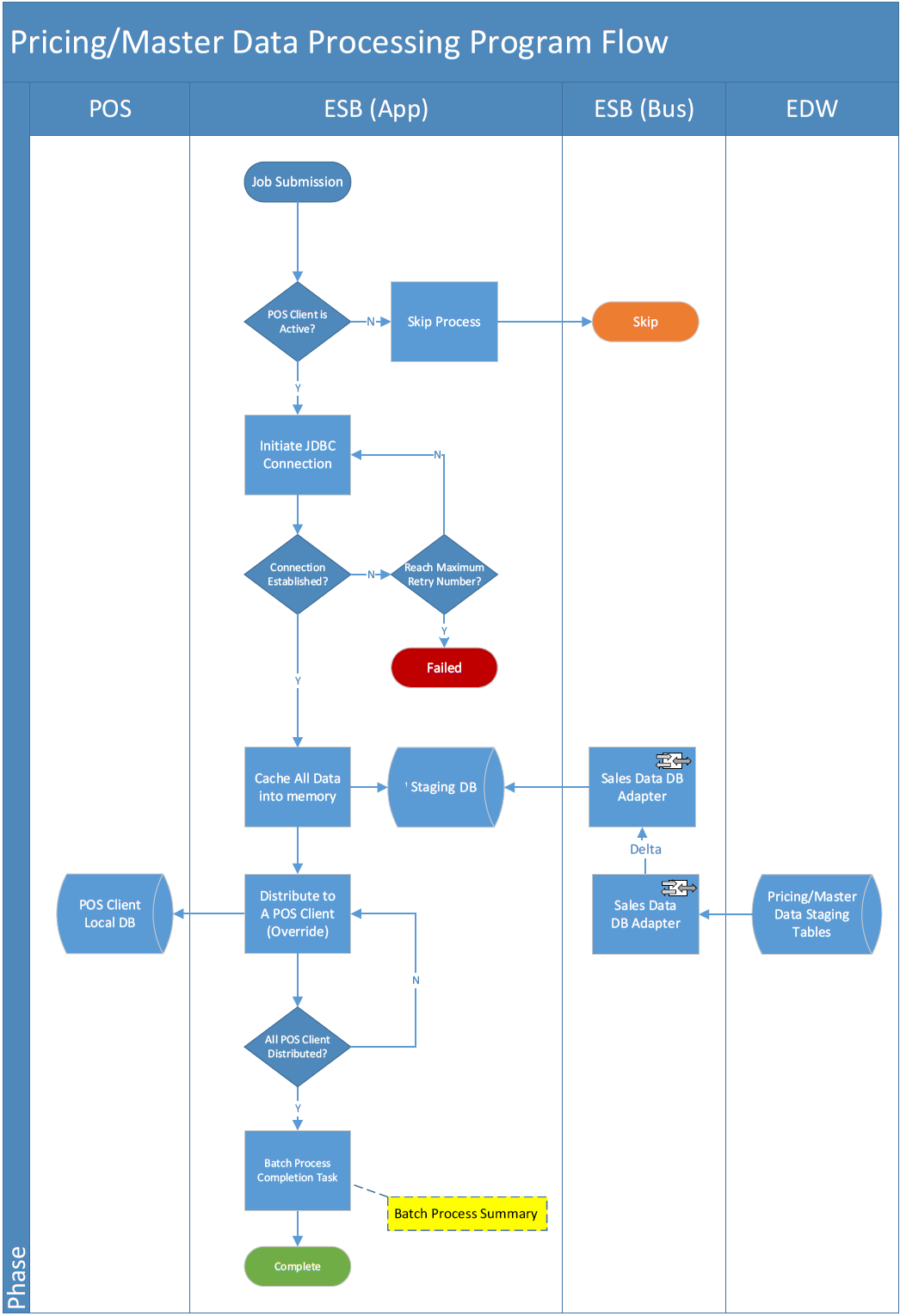
| Application | Description |
| --- | --- |
| Polling Gateway | The application module responsible of downloading data from POS client. |
| Sending Gateway | The application module to carry out data pushing to service bus interfaces. |
| ESB Interfaces | The interface for data pushing from the Staging DB of service bus application layer 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 successfully sent to EDW, the sending result shall revert 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.  Pricing/Master data will distribute to POS client in CSV format. (2 types of csv file). File format information to be provided by Maxim’s IT. | Output | Sales Data Download to Staging DB – Carl |
| Distribution job completion | Log down job status and process summary of the POS client for record tracking. | 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 the right to conduct job re-run, job logs and dashboard; demand of daily summary report. | Operator |
| POS Support | The user role has the 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 the Staging DB of service bus application layer 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 submitted by the job controller initiates a JDBC connection to update data into POS client. | Must Have | Carl | Assume the pricing/master data have the primary keys, the processing will update/merge the data by the reference keys 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 is no reference of the data indicating it is 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.  Transaction commitment control provided to avoid partially updated records. | 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 data in POS clients and merge any changes continually. | 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 successfully input into POS client, the sending result shall revert 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  Job run status  Number of records processed  Job name, e.g. Master – Staging – POS  Job ID  Job run error stage (if any)  Error Category, e.g. DB, Network and Other, etc.  Job start/end time (regarded as Coping to POS start/end time) | ESB-MST-REQ-02 |

## Non-functional Requirements

### 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 dashboard 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’s IT | Initial design should contain information below for the dashboard purpose:   1. Because there are over 400 clients, so a combo box menu will allow user to select one batch, 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 information:    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’s 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 7x24, 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’s 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 the exception’s severity. | Must Have | Maxim’s 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 categorized by 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’s IT | The scenarios of data exception:   1. Data not match 2. The data cannot complete an entity relationship (e.g. order, trans, 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’s 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 introduced into several stages   * EOD processing | Nice to have | Maxim’s IT | EOD process will allow user to trigger re-run (UI based), when encounter job failure, the system admin will manually 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’s 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’s existing system called “Branch Master” | Nice to have | Maxim’s IT | The service bus control of the POS client branch information will initiate from “branch master” 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’s IT | Critical System Parameters will be stored 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 EOD. | Must Have | Maxim’s 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’s IT | Pricing/Master data 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’s 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’s 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’s IT | After the EOD data transferred to EDW if total record count and total amount have been validated, update the convert log in EDW to indicate 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: 4.2.3, 4.2.5, 4.3.4) | | |
| Name (Print or Type) | Date | Signature |
| CARL CHOW |  |  |
|  | | |
| Staging to EDW Part Sign-Off (4.2.4, 4.2.6) | | |
| Name (Print or Type) | Date | Signature |
| POLLY KAM |  |  |
|  | | |
| Others Sections Sign-Off (Others) | | |
| Name (Print or Type) | Date | Signature |
| CHOI KA WING |  |  |
|  | | |
| Project Director | | |
| Name (Print or Type) | Date | Signature |
| LOUIS MAH |  |  |
|  | | |

# Appendix A – Existing Polling Servers Overview



# Appendix B – Data Process Flowchart



# Appendix C – EDW Virtual Branch Sales Inbound Tables



# Appendix D – POS Client Polling Table

| ***Table*** | ***Type*** | ***Update Freq.*** | ***Up/Down*** | ***Source*** |
| --- | --- | --- | --- | --- |
| ACCOUNTS | Master | Every Polling | Download | Carl |
| COUPON\_CONTROL | Master | Every Polling | Download | Carl |
| COUPON\_RANGE | Master | Every Polling | Download | Carl |
| CURRENCY | Master | Every Polling | Download | Carl |
| EMPLOYEE | Master | Every Polling | Download | Carl |
| HIST\_CHECK\_LOGS | HIST | EOD | Upload | Carl |
| HIST\_COUPON\_SALES | HIST | EOD | Upload | Carl |
| HIST\_ITEM | HIST | EOD | Upload | Carl |
| HIST\_ITEMSTOCK | HIST | EOD | Upload | Carl |
| HIST\_ORDERS | HIST | EOD | Upload | Carl |
| HIST\_ORDERS\_EXTRA | HIST | EOD | Upload | Carl |
| HIST\_ORDERS\_PAY | HIST | EOD | Upload | Carl |
| HIST\_ORDERS\_PAY\_PROGRESS | HIST | EOD | Upload | Carl |
| HIST\_PAYFIG | HIST | EOD | Upload | Carl |
| HIST\_PAYSUM | HIST | EOD | Upload | Carl |
| HIST\_POSSYSTEM | HIST | EOD | Upload | Carl |
| HIST\_REDEEMED\_COUPON | HIST | EOD | Upload | Carl |
| HIST\_SAFEBOXCHECK | HIST | EOD | Upload | Carl |
| HIST\_SAFEBOXCHECKTENDER | HIST | EOD | Upload | Carl |
| HIST\_SAFEBOXINOUT | HIST | EOD | Upload | Carl |
| HIST\_SAFEBOXINOUTEXTENDINFO | HIST | EOD | Upload | Carl |
| HIST\_SAFEBOXPICKUP | HIST | EOD | Upload | Carl |
| HIST\_SESSIONINFO | HIST | EOD | Upload | Carl |
| HIST\_SESSIONTENDER | HIST | EOD | Upload | Carl |
| HIST\_STOCK\_MOVEMENT | HIST | EOD | Upload | Carl |
| HIST\_SUPP | HIST | EOD | Upload | Carl |
| HIST\_TRANS | HIST | EOD | Upload | Carl |
| HIST\_TRANS\_ECARD | HIST | EOD | Upload | Carl |
| HIST\_TRANS\_MODIFIER | HIST | EOD | Upload | Carl |
| INVITATION | Master | Every Polling | Download | Carl |
| ITEM | Master | Every Polling | Download | Carl |
| ITEM\_BARCODE | Master | Every Polling | Download | Carl |
| ITEM\_MODIFIER | Master | Every Polling | Download | Carl |
| ITEMANLY | Master | Every Polling | Download | Carl |
| ITEMDEPT | Master | Every Polling | Download | Carl |
| MENU | Master | Every Polling | Download | Carl |
| MENUITEM | Master | Every Polling | Download | Carl |
| MESSAGES | Master | Every Polling | Download | Carl |
| MODIFIER | Master | Every Polling | Download | Carl |
| MODIFIER\_GRP | Master | Every Polling | Download | Carl |
| MODIFIER\_LIST | Master | Every Polling | Download | Carl |
| ONHOUSE | Master | Every Polling | Download | Carl |
| OPTIONS | Master | Every Polling | Download | Carl |
| ORDERS | Sales Data | Every Polling | Upload | Carl |
| ORDERS\_EXTRA | Sales Data | Every Polling | Upload | Carl |
| ORDERS\_PAY | Sales Data | Every Polling | Upload | Carl |
| PAYCAT | Master | Every Polling | Download | Carl |
| PAYMENT | Master | Every Polling | Download | Carl |
| PMT\_ACTION | Master | Every Polling | Download | Carl |
| PMT\_CONDITION | Master | Every Polling | Download | Carl |
| PMT\_HDR | Master | Every Polling | Download | Carl |
| POSBUSDATE | Master | Every Polling | Download | Carl |
| ROLE | Master | Every Polling | Download | Carl |
| ROLE\_PERMISSION | Master | Every Polling | Download | Carl |
| SUPP | Sales Data | Every Polling | Upload | Carl |
| SYSSETTINGS | Master | Every Polling | Download | Carl |
| TRANS | Sales Data | Every Polling | Upload | Carl |
| TRANS\_ECARD | Sales Data | Every Polling | Upload | Carl |
| TRANS\_MODIFIER | Sales Data | Every Polling | Upload | Carl |
| TRANS\_TYPE | Master | Every Polling | Download | Carl |
| USER\_PERMISSION | Master | Every Polling | Download | Carl |
| USER\_ROLE | Master | Every Polling | Download | Carl |
| WIFI CODE | Master | Every Polling | Download | Carl |

# Appendix E – EDW Tables

|  |  |  |  |
| --- | --- | --- | --- |
| **Tables** | **Source** | **Matched with Polling Tables** | **Remark** |
| HIST\_COUPON\_SALES | EDW | HIST\_COUPON\_SALES | EOD |
| HIST\_ITEM | EDW | HIST\_ITEM | EOD |
| HIST\_ITEMSTOCK | EDW | HIST\_ITEMSTOCK | EOD |
| HIST\_ORDERS | EDW | HIST\_ORDERS | EOD |
| HIST\_ORDERS\_PAY | EDW | HIST\_ORDERS\_PAY | EOD |
| HIST\_PAYFIG | EDW | HIST\_PAYFIG | EOD |
| HIST\_PAYSUM | EDW | HIST\_PAYSUM | EOD |
| HIST\_POSSYSTEM | EDW | HIST\_POSSYSTEM | EOD |
| HIST\_REDEEMED\_COUPON | EDW | HIST\_REDEEMED\_COUPON | EOD |
| HIST\_SUPP | EDW | HIST\_SUPP | EOD |
| HIST\_TRANS | EDW | HIST\_TRANS | EOD |
| HIST\_TRANS\_ECARD | EDW | HIST\_TRANS\_ECARD | EOD |
| HIST\_TRANS\_MODIFIER | EDW | HIST\_TRANS\_MODIFIER | EOD |
| ITEM | EDW | ITEM | Master |
| TRANS\_TYPE | EDW | TRANS\_TYPE | Master |
| COUPON\_SALES | EDW | COUPON\_SALES | Real Time |
| ITEMSTOCK | EDW | ITEMSTOCK | Real Time |
| ORDERS | EDW | ORDERS | Real Time |
| ORDERS\_PAY | EDW | ORDERS\_PAY | Real Time |
| SUPP | EDW | SUPP | Real Time |
| TRANS | EDW | TRANS | Real Time |
| TRANS\_ECARD | EDW | TRANS\_ECARD | Real Time |
| TRANS\_MODIFIER | EDW | TRANS\_MODIFIER | Real Time |
| CAKE\_ORDERS | EDW | CAKE\_ORDERS (IT52 only) | Not in scope (for ref. only) |
| CAKE\_TRANS | EDW | CAKE\_TRANS (IT52 only) | Not in scope (for ref. only) |
| EVENT\_ORDERS | EDW | EVENT\_ORDERS (IT51 only) | Not in scope (for ref. only) |
| EVENT\_TRANS | EDW | EVENT\_TRANS (IT51 only) | Not in scope (for ref. only) |
| HIST\_TRANSSUMMARY | EDW | HIST\_TRANSSUMMARY | Not in scope (for ref. only) |
| HIST\_VOIDORDERS | EDW | HIST\_VOIDORDERS | Not in scope (for ref. only) |
| HIST\_VOIDTRANS | EDW | HIST\_VOIDTRANS | Not in scope (for ref. only) |
| VIEW\_HIST\_TRANS\_PAST\_ONLY | EDW | VIEW\_HIST\_TRANS\_PAST\_ONLY | Not in scope (for ref. only) |
| EMFC\_TRANS\_LOG | EDW | EMFC\_TRANS\_LOG | Not in scope (for ref. only) |
| CONVERT\_LOG | EDW | CONVERT\_LOG | Not in scope (for ref. only) |

# Appendix F - Report Templates

|  |  |
| --- | --- |
|  | 未能上數分店數量 |
| 香港美心西餅 | 3 |
| 東海堂西餅 | 3 |
| 廣州Mei-Xin西餅 | 4 |
| 總數 | 10 |

| Convert Count | | Green: 5%  Yellow: 5-10%   Red: >10% | |
| --- | --- | --- | --- |
| **BU** | Time | Converted  (color of this column will be eliminated) | Expected |
| **CAK** | 9:00pm | 57 | 44 |
| 10:00pm | 122 | 94 |
| 11:00pm | 194 | 133 |
| 12:00am | 249 | 167 |
| 01:00am | 290 | 254 |
| 02:00am | 331 | 311 |
| 03:00am | 332 | 350 |
| 04:00am | 338 | 350 |
| 07:00am | 340 | 350 |
| **CHI** | 9:00pm | 1 | 1 |
| 10:00pm | 2 | 1 |
| 11:00pm | 5 | 2 |
| 12:00am | 17 | 4 |
| 01:00am | 44 | 4 |
| 02:00am | 59 | 14 |
| 03:00am | 60 | 61 |
| 04:00am | 60 | 61 |
| 07:00am | 60 | 61 |
| **EUR** | 9:00pm | 3 | 2 |
| 10:00pm | 5 | 6 |
| 11:00pm | 14 | 8 |
| 12:00am | 26 | 11 |
| 01:00am | 34 | 19 |
| 02:00am | 37 | 29 |
| 03:00am | 42 | 37 |
| 04:00am | 42 | 43 |
| 07:00am | 42 | 43 |
| **FFS** | 9:00pm | 29 | 28 |
| 10:00pm | 52 | 46 |
| 11:00pm | 81 | 103 |
| 12:00am | 116 | 125 |
| 01:00am | 125 | 129 |
| 02:00am | 125 | 129 |
| 03:00am | 135 | 136 |
| 04:00am | 135 | 138 |
| 07:00am | 135 | 138 |
| **JCR** | 9:00pm | 1 | 1 |
| 10:00pm | 4 | 4 |
| 11:00pm | 4 | 6 |
| 12:00am | 21 | 75 |
| 01:00am | 154 | 147 |
| 02:00am | 164 | 159 |
| 03:00am | 166 | 164 |
| 04:00am | 168 | 167 |
| 07:00am | 168 | 167 |
| **SBS** | 9:00pm | 11 | 16 |
| 10:00pm | 16 | 37 |
| 11:00pm | 33 | 55 |
| 12:00am | 58 | 87 |
| 01:00am | 103 | 120 |
| 02:00am | 113 | 162 |
| 03:00am | 152 | 168 |
| 04:00am | 159 | 171 |
| 07:00am | 160 | 171 |

Error Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Server**  **/ Job name** | **Check Time**  **Failed**  **/Stall** |  | **Action** |
| udj\_Gen\_Pricing (HOPOS) | 11/30/2016 9:47:55 PM  Is failed | High Level Grouped Error Message such as:   * Change POS type (e.g. change from Qualicom to PointSoft) * Password Setting in POS is incorrect and cannot extract data * Network issue, ping failed * Store did not perform day end process, need to split sales data * Imbalance (missing data file/incomplete data files from POS machine) * Invalid branch (not in our branch master/invalid length) -> data can be loaded to Oracle staging but convert log show “error” until POS vendor fix the issue. * Invalid item code (not in our item master/invalid length) -> data can be loaded to Oracle staging but convert log show “error” until POS vendor fix the issue. | 11/30/2016 10:10:00 PM  Rerun  11/30/2016 10:20:00 PM  Call SA  11/30/2016 11:30:00 PM  succeeded |

**Missed Branch (merged into error log)**

|  | Branch | Name | Day end | Client Server | ping |
| --- | --- | --- | --- | --- | --- |
| GEN | 6156 | 6156 | 2016/12/1 0:02 | M6156 | 0%loss |
| SBS | 4364 | 北衛星客運廊 Starbucks | 2016/11/30 23:11 | M4364\_01 | 0%loss |
| EUR | 1298 |  | 2016/11/30 22:40 |  |  |
| SBS | 4609 | Pacific Place | 2016/11/30 21:50 | m4609\_01 | 0%loss |
| SBS | 4432 | Three Exchange Square | 2016/11/30 21:47 | M4432\_01 | 0%loss |
| SBS | 4302 | Star Ferry | 2016/11/30 21:39 | M4302\_01 | 0%loss |
| SBS | 4626 | MTR - Central | 2016/11/30 20:59 | m4626\_01 | 0%loss |
| FFS | 4692 | Starbucks 寫字樓 | 2016/11/30 18:22 | ITPOSTS | 0%loss |
| SBS | 4683 | CORPORATE SALES | 2016/11/30 17:49 | ITPOSTS | 0%loss |
| CAKE | 3745 | GZMTR- 芳村 | 2016/11/30 21:54 | M3745\_01 | 0%loss |
| CAKE | 3476 | 金鐘站餅店 | 2016/11/30 21:38 | M3476\_01 | 0%loss |
| CAKE | 5510 | 香港仔(東海堂) | 2016/11/30 21:09 | M5510\_01 | 0%loss |
| CAKE | 3133 | 竹園餅店 | 2016/11/30 20:21 | M3133\_01 | 0%loss |
| GEN | 2822 | 2822 | 2016/12/1 2:50 | M2822 | 0%loss |
| FFS | 2222 | 杏花MX | 2016/11/30 21:54 | M2222 | 0%loss |
| ICD | 2881 | 瑞信銀行職員餐廳 | 2016/11/30 19:34 | M2881 | 0%loss |
|  |  |  |  |  |  |

\*Figures appear in the reports shown in this section can be exported.

- End -