

High Level Requirement

Optimus Core: Order Processing

# Contents

[1 Contents 2](#_Toc398899242)

[2 Executive Summary 3](#_Toc398899243)

[2.1 Purpose of document 3](#_Toc398899244)

[2.2 High level outline of project and business benefit 3](#_Toc398899245)

[3 Document Control 4](#_Toc398899246)

[3.1 Versioning 4](#_Toc398899247)

[3.2 Reviewing information 4](#_Toc398899248)

[3.3 Review history 5](#_Toc398899249)

[4 Requirement Specifications: Generic features 5](#_Toc398899250)

[4.1 Customer Service Interface 5](#_Toc398899251)

[4.2 Order Search & View 5](#_Toc398899252)

[4.3 Order Approval Flow 6](#_Toc398899253)

[4.4 Order History 6](#_Toc398899254)

[4.5 Payment Handling 7](#_Toc398899255)

[4.6 Send Status Notifications to a Separate System 7](#_Toc398899256)

[4.7 Reporting 8](#_Toc398899257)

[5 Requirement Specifications: Delivery 8](#_Toc398899258)

[5.1 Basic Delivery Options 8](#_Toc398899259)

[6 Requirement Specifications: Order processing features (Workflow steps) 9](#_Toc398899260)

[6.1 Add order 9](#_Toc398899261)

[6.2 Send for Fraud Check 10](#_Toc398899262)

[6.3 Allocate stock – link to Stock Orchestration module 10](#_Toc398899263)

[6.4 Order awaiting approval 11](#_Toc398899264)

[6.5 Manage back orders 11](#_Toc398899265)

[6.6 Take payment 12](#_Toc398899266)

[6.7 Order Cancellation 12](#_Toc398899267)

[6.8 Refund payment 13](#_Toc398899268)

[6.9 Send to warehouse (i.e. ERP) 14](#_Toc398899269)

[6.10 Check logistics status (i.e. warehouse and shipping status) 14](#_Toc398899270)

[6.11 Vouchers/ Gift cards/ Promotions 15](#_Toc398899271)

[7 Requirement Specifications: Returns 15](#_Toc398899272)

[8 Requirement Specifications: Integrations 15](#_Toc398899273)

[8.1 Product Catalogue application 15](#_Toc398899274)

[8.2 Stock Orchestration Module 16](#_Toc398899275)

[8.3 Picasso Payment Solution Integration (including credit cards, debit cards & 3D Secure) 16](#_Toc398899276)

[8.4 ERP integration 16](#_Toc398899277)

[8.5 Comms Module 17](#_Toc398899278)

[9 Requirement Specifications: B2B 17](#_Toc398899279)

[10 Requirement Specifications: Other services 17](#_Toc398899280)

[11 Requirement Specification: Generic 17](#_Toc398899281)

[11.1 Generic requirements 17](#_Toc398899282)

[12 Glossary 18](#_Toc398899283)

[13 Data Confidentiality Statement for this document 19](#_Toc398899284)

[13.1 Confidentiality Policy 19](#_Toc398899285)

[13.2 Intended audience 20](#_Toc398899286)

[13.3 Intended audience audit 20](#_Toc398899287)

# Executive Summary

## Purpose of document

This document contains the requirements for the Optimus Order Processing project, the requirements will be outlined from a business perspective and will not contain any technical details; this document will enable a clear understanding of the requirements to be obtained by all of the project team, and therefore enable a project impact assessment to be completed, and the project to be estimated, but it won’t contain all of the detailed requirements for the solution – these will be discussed and agreed during the development phase of the project.

Note: only the phase 1 requirements are detailed.

## High level outline of project and business benefit

The Optimus Order Processing application enables the implementation of a client specific Order Processing application which receives orders from all channels.

The Order Processing application provides the functionality to the Order Process Administrators to enable them to manage the necessary activities for an order that has been created until the point where the order is ready for it to be sent to the ERP ready for the pick and pack procedures to be completed.

There are 2 main types’ of admin interfaces for the Order Processing application:

1. For the System Administrator to configure aspects of the system.
2. For the Order Processing Administrators to complete the necessary activity on an order (as per the order workflow step business rules).

Figure 1: High level diagram of the Order Processing application

# Document Control

## Versioning

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ver. # | Change Description | Release Reference | Date | Author |
| 0.1 | Implementing phase 1 of Optimus Order Processing | 1.0 | 19/9/14 | Sarah Toogood |
| 0.2 | Feedback from OM, PI, MK, VB, MG | 1.0 | 2/10/14 | Sarah Toogood |

## Reviewing information

|  |  |
| --- | --- |
| Document Owner (Business Analyst) | Sarah Toogood |
| Technical Owner & Product Owner | Piotr Ignaczak |
| Infrastructure Owner | Jake van Schaik |
| Software Architecture Owner | Piotr Ignaczak |
| Program Manager | Ozkan Mustafa |

## Review history

|  |  |  |
| --- | --- | --- |
| Document Ver. # | Reviewers | Sign off status |
|  |  |  |

# Requirement Specifications: Generic features

## Customer Service Interface

### Requirement statement

The Order Processing application enables an administrator to access the data within the system to see the necessary information and manage orders and the system as required.

### Requirement scenarios

1. There are four main types of users’ in the system are:
   1. Order read only user
   2. Order approvals user
   3. Account administrator
   4. System administrator (for several accounts) e.g. for the implementation team to use.
2. The order read only user can log into the system and see orders for the account(s) they have access to. They can see the data but not make any amendments.
3. The order approvals user can log into the system and see orders for the account(s) they have access to, and add details and approve or decline orders etc.
4. The account administrator can configure the settings for one specific account e.g. add users or change workflows.
5. The system administrator can configure the settings for several accounts e.g. add users or change workflows.
6. The screens available for the customer’s administration screen to view orders and approved orders are:
   1. Order search screen
   2. Order approval screen
   3. Order details
7. The screens available for the account and system administrator:
   1. System configuration screen
   2. User access

## Order Search & View

### Requirement statement

An order user can view and search orders within the system.

### Requirement scenarios

1. The search options available are:
   1. Order status
   2. Order reference number
   3. Customer details
   4. Product details
   5. Order created date
2. The user can combine the different search options e.g. status and product details.
3. From the search results the user can access the order details screen, and complete any necessary activity (dependent upon the order status).

## Order Approval Flow

### Requirement statement

The Order Processing application supports a flexible workflow for an order, i.e. the order steps available within the system can be configured to be in a specific order for a client or channel.

### Requirement scenarios

1. The order processing can support a series of workflow steps which are used to manage orders. The workflow order can be configured to be in any order for a specific client or specific channel e.g. check order occurs before fraud check etc.
2. In phase 1 there is no administrator screen to support the configuring of the workflow order.
3. Workflow steps available are:
   1. Add order
   2. Validate order/quarantine order
   3. Fraud check – link to Fraud Management application
   4. Manage back orders
   5. Allocate stock – link to Stock Orchestration module
   6. Payment handling:
      1. Take payment
      2. Refund payment
   7. Order cancellation
   8. Send to warehouse (i.e. ERP)
   9. Check logistics status (i.e. warehouse and shipping status)
   10. Return order – not supported in phase 1.
   11. Order amendment – not supported in phase 1.
   12. Note: other workflow steps will be added in later steps.

## Order History

### Requirement statement

A user can see the following historical information against an order:

* Status history
* Notes history
* Notification history
* Order amendment history – not in phase 1.

### Requirement scenarios

1. On the order detail screen the administrator can see the following historical information against an order:
   1. Status history
   2. Notes history
   3. Notification history
2. Status history shows all of the changes to the status (e.g. status name, date of the change, and who changed the status of an order).
3. The notes history shows all notes against an order (e.g. note, user and date).
4. The notification history shows all the automated information about emails against an order (e.g. order confirmation). Note: only emails are supported in phase 1.

## Payment Handling

### Requirement statement

The Order Processing Application can manage payments against an order. In phase 1 only credit card and debit card payments are supported.

### Requirement scenarios

1. The Order Processing Application supports:
   1. Taking payments
   2. Refunding payments
   3. Cancelling payments (i.e. pre-authorisation)
2. The payment methods supported are:
   1. Credit cards and debit cards only in phase 1.
3. For credit cards and debit cards the core supports two taking payment flows:
   1. Pre-auth and fulfilment payments taken at the relevant stage.
   2. OR taking fulfilment payment only.
4. The Order Processing application supports a client implementation which can either take credit/debit card payment via:
   1. Integration with the Picasso Payment Solution.
   2. Integration with a payment gateway which is on a separate site and page e.g. Paysbuy for dtac. This would be done via an extension, but the core will support this.

## Send Status Notifications to a Separate System

### Requirement statement

The Order Processing application will send status updates via the notification API to external systems e.g. eCommerce core, Customer System etc. This just informs the other system that an update to the status has occurred (it is a passive notification), this could be needed to provide information to the end customer through a User Interface; no additional information is expected from the other system through this process, just notification that the status update has been received.

### Requirement scenarios

1. Notification is sent for all status changes, it can be switched on or off for all statuses.
2. The API will push the information to the other system, in an asynchronous manner, i.e. whilst the status updates could be near to real-time, it might not be real-time depending upon the queue of updates to send.
3. The API is a standard API format provided, if a client requires a specific API format this would be developed through an extension.

## Reporting

### Requirement statement

The Order Processing application will make available key order information to the reporting tool as part of its workflow steps.

### Requirement scenarios

1. Key order information will be made available to the reporting tool, via the Middleware.
2. There is no operational dashboard within the Order Processing application in phase 1.

# Requirement Specifications: Delivery

## Basic Delivery Options

### Requirement statement

The Order Processing application will support the basic delivery option in phase 1, i.e. delivery from a warehouse or a customer’s address (this can be business or personal) via a courier.

Additional delivery options will be added in later phases.

### Requirement scenarios

1. The delivery options will support various delivery services e.g. standard, tracked, next day etc.
2. The delivery option will be specified against an order when the order is initially received into the system.
3. The delivery details will be specified within the Product Catalogue application, or linked to the customer’s Product System.

# Requirement Specifications: Order processing features (Workflow steps)

The default high level workflow is as follows:



Figure 2: High level default order workflow steps

## Add order

### Requirement statement

The Order Processing system supports an order being added from any system, as part of this process the order information is validated.

### Requirement scenarios

1. The integration with the Order Process application is through a standard API; if a different API is required this would be managed by an extension. The systems which could generate an order include the following systems:
   1. eCommerce Core
   2. ePos
   3. Client 3rd system
2. In phase 1 an order cannot be manually added to the Order Processing application (e.g. by a Customer Service agent or by a telesales agent).
3. When the order details are received the Order Processing application automatically validates:
   1. Key order information is listed – e.g. client account, sales channel, delivery address, billing address, basket information (including Bundles & vouchers) etc.
   2. Products – do they exist within the Product Catalogue application (or relevant client Product System)?
   3. Delivery type – does it match one listed within the Product Catalogue application (or relevant client Product System)?
4. If an order fails basic validation the order is not added to the Order Processing application and a rejection response is sent to the originating system.
5. The next step for an order is:
   1. Order rejected, and notification sent to originating system
   2. Next order workflow step
6. Link to stock orchestration module for awareness – need to confirm when stock allocation module is defined.

## Send for Fraud Check

### Requirement statement

The Order Processing system can send an order to the Fraud Management application for a fraud check to be carried out.

### Requirement scenarios

1. At the configured point in the process the order is sent to the Fraud Management application (if purchased by the client) to be fraud checked.
2. The results of the fraud decision received from the Fraud Management application are:
   1. Automatically declined
   2. Automatically approved
   3. Manually declined
   4. Manually approved
3. As a result of the above statuses the result of this order step is:
   1. Next step in the workflow e.g. allocate stock (approved)
   2. Cancel order (declined).

## Allocate stock – link to Stock Orchestration module

### Requirement statement

The Order Processing system enables an order to have stock allocated to it, this is done using the information from the Stock Orchestration module.

### Requirement scenarios

1. At the configured point in the process the order is ready for the stock to be allocated for each physical product in the order.
2. This step is only completed for physical products in the order in phase 1.
3. Using information from the stock orchestration module the Order Processing application will know whether there is sufficient available stock in the warehouse to complete this whole order.
4. As a result of the above check the result of this order step is:
   1. Next step in the workflow e.g. Take payment
   2. Manage back order

## Order awaiting approval

### Requirement statement

The Order Processing system provides an interface for the Order Processing administrator to validate orders, according to their business rules, e.g. a manual screen to manage an order.

### Requirement scenarios

1. At the configured point in the process the order is sent to an order awaiting approval order workflow step.
2. The screen will show all orders which are in awaiting approval.
3. The user can add notes against the order e.g. details of investigation or information from the customer. All notes are audited.
4. The user can view detailed information on the order.
5. The administrator can move an order to:
   1. Quarantine – stay in this workflow but with a different status to identify them as having a problem.
   2. Declined (Cancelled)
   3. Next workflow step (can be specified).
6. The user can also access a screen to search for all orders (even those already fraud check completed, or those being fraud checked).

## Manage back orders

### Requirement statement

The Order Processing application will enable an Order Administrator to manage any orders which do not have enough stock in the warehouse to fulfil.

### Requirement scenarios

1. An Order Administrator can see a list of all orders which could not be have stock allocated as part of the ‘Allocate Stock’ check.
2. The Order Administrator can add a note against the order e.g. result of investigation on why there is no stock, update from customer’s decision, information from the Product Manager or supplier etc.
3. For order in this status the administration can choose to:
   1. Cancel order
   2. Wait for stock to arrive
4. The Order Processing application regularly uses information from the stock orchestration module the Order Processing application will know whether there is sufficient available stock in the warehouse to complete this whole order.
5. The Order Processing application prioritises orders in a back order status to have stock allocated over the other orders in the system by order created date.
6. As a result of the above check the result of this order step is:
   1. Next step in the workflow e.g. Take payment
   2. Manage back order
   3. Cancel order
7. Part shipments and order amendments are not part of phase 1 delivery.
8. Add information on how get updates on new stock levels from Stock Orchestration module.

## Take payment

### Requirement statement

The Order Processing application can manage two methods of receiving payment via credit or debit card. The full payment can be taken at the configured workflow order step, or a pre-authorisation can be taken and then at the configured workflow order step the full payment is taken.

### Requirement scenarios

1. The client’s order workflow can be set up to either take pre-authorisation and then full payment or take just a full payment.
2. For the scenario when a pre-authorisation would be taken for the order is added to the Order Processing application; the order details sent to the Order Processing application include a payment reference id. The Order Processing application then takes full payment (using the payment reference id) at the configured workflow step.
3. The Order Administrator team can see the status of a payment e.g. pending, failed, rejected or completed.
4. The Order Administrator can add a note against the order e.g. payment has been verified as successful manually by bank transfer, notes from contact with the customer, etc.
5. For a failed or rejected payment the Order Administrator can:
   1. Cancel an order
   2. Request for a payment to be retried; which retriggers this workflow.
   3. Mark as payment completed, which triggers the next step in the order workflow.
6. In the case of a full payment being taken this would be taken before the order is added to the system, as this is necessary for adhere to PCI DSS (the Order Processing application will not take or store credit card information); in this case the order includes a payment reference id. For payments which failed to be taken as part of this process they would not be added into the system; for any delayed notification on payments of this type a client extension would be created.

## Order Cancellation

### Requirement statement

A user can cancel an order (dependent upon specific business rules), when an order is cancelled if necessary it will trigger a refund payment, as well as sending notification to the customer. If necessary it will also update stock levels within the stock orchestration module.

### Requirement journey

Figure 3 – High level process to cancel an order

### Requirement scenarios

1. The workflow will determine when an order is allowed to be cancelled e.g. if an order has been shipped it cannot be cancelled.
2. In phase 1, the workflow for cancelling an order is fixed, in later phases additional workflow configurability will be added to support different scenarios with ERP and alternative payment methods.
3. The order is validated, both within the Order Processing application as well as, if necessary, the ERP system to determine whether the order meets the configured business rules to accept cancellation e.g. order is not being pick and packed or shipped.
4. The integration with the ERP is through a standard API, if a different API is required this would be managed by an extension. The ERP support is either the Optimus ERP, or a client specific ERP.
5. If necessary a refund request is triggered, this is managed through a separate workflow step.
6. If configure the cancellation workflow will trigger a request to send an email to a customer to notify them of the confirmed cancellation. This communication is sent from the Comms Module.
7. Dependent upon whether the order was submitted to the ERP at the point it is cancelled will depend upon whether an update in the stock levels needs to be sent to the stock orchestration module. If the order has not been submitted to the ERP then the stock orchestration module stock levels will be updated; if the order was submitted to the ERP the ERP would update the stock levels in the stock orchestration module.

## Refund payment

### Requirement statement

The Order Processing application can refund a payment to a credit or debit card as part of the cancellation workflow in phase 1. (In future phases this would include be triggered by a gesture of goodwill, and returns order workflow steps)

### Requirement scenarios

1. Refunds are made against the payment reference id. The technical method for refunding the payment will depend upon the client’s implementation.
2. If a failed or rejected refund then the administrator can see the list of orders and complete the following steps:
   1. Request for a refund to be retried; which retriggers this workflow.
   2. Mark as refund completed, which triggers the next step in the order workflow.
3. The Order Administrator team can see the status of a refund e.g. pending, failed, rejected or completed.
4. The Order Administrator can add a note against the order e.g. refund has been verified as successful manually by bank transfer, notes from contact with the customer, etc.
5. Details of the refund are logged within the audit.

## Send to warehouse (i.e. ERP)

### Requirement statement

The Order Processing application can send an order to the ERP application for the order to be fulfilled and delivered to the customer.

### Requirement scenarios

1. At the configured point in the process the order is sent to the ERP application (if purchased by the client) to be fulfilled.
2. A response from the ERP is expected to confirm whether the order has been accepted or rejected.
3. If the order has been rejected then the Order Administrator can trigger the order to be resent to the ERP.
4. An Order Administrator can see a list of orders which have been sent to the ERP.
5. The order details sent to the ERP are through a standard API; if a client requires a different format then this would be managed through a client extension.
6. If an order has no stock available in order to fulfill it then the order is rejected by the ERP; thus back orders are only managed in the Order Processing application.

## Check logistics status (i.e. warehouse and shipping status)

### Requirement statement

The Order Processing application will query the ERP system to receive status updates so that this information can be displayed to an order administrator (and also fed into other systems e.g. the eCommerce core to display to the end customer).

### Requirement scenarios

1. This step must occur after the order has been sent to the ERP.
2. The request for the order status update is through a standard API; if a client requires a different format then this would be managed through a client extension.
3. The status of an order in the Order Processing application are:
   1. Pending dispatch
   2. Being processed i.e. picked and packed
   3. Ready for shipment
   4. Cancelled
   5. On hold
   6. Note: a client’s ERP statuses would be mapped to these Order Processing application statuses through a client extension.
4. Status updates would be requested periodically until an order is in the ready for shipment status.
5. Note: Delivery statuses are not covered in phase 1 as there is not integration with the CBP.
6. If an order is marked as cancelled within the ERP, this would trigger the cancellation workflow in the Order Processing application.
7. The ready for shipment status could be used to trigger a workflow order step e.g. take payment.

## Vouchers/ Gift cards/ Promotions

### Requirement statement

The Order Processing application supports orders which have a voucher included.

### Requirement scenarios

1. Any voucher information would need to be sent to the ERP as part of the sent to warehouse – this would ensure that the information is in the necessary finance system.
2. In phase 1 vouchers will only support money off or a free product. (Note: free products are not supported by the eCommerce core in phase 1) (Later phase enhancements will include adding a product, gift cards, buy one get one free etc.).
3. A voucher can be at a product level, a product bundle level or an order level; all of these are supported by the Order Processing application.
4. Gift cards and other promotion types are not supported in phase 1.

# Requirement Specifications: Returns

The management of returns through the order processing system will not be supported in phase 1.

# Requirement Specifications: Integrations

## Product Catalogue application

### Requirement statement

The Order Processing application is integrated with the Optimus Product Catalogue application; the integration enables the Order Processing application to obtain information regarding products, product prices, delivery options, vouchers and bundles.

### Requirement scenarios

1. At the appropriate workflow step information is pulled from the Product Catalogue by the Order Processing application for product, product prices, delivery options, vouchers and bundle for the information on an order.
2. There is a standard format for the integration.
3. If the client requires integration with a client specific Product System this would be managed through an extension.

## Stock Orchestration Module

### Requirement statement

The Order Processing application is integrated with the Stock Orchestration Module; the integration enables the Order Processing application to obtain information on stock levels, and allocate stock to an order.

### Requirement scenarios

1. At the appropriate workflow steps information is retrieved or request sent to the Stock Orchestration module to enable stock levels to be checked for an order, and stock to be allocated to an order.
2. There is a standard format for the integration.
3. This would include workflow steps for allocating stock as well as managing back orders.
4. If the client requires integration with a client specific Stock Level system this would be managed through an extension.

## Picasso Payment Solution Integration (including credit cards, debit cards & 3D Secure)

### Requirement statement

The Order Processing application enables each client implementation to optionally be integrated with the Picasso Payment Solution (enabling PCI DSS adherence), and the associated payment gateways integrated with this product.

For clients who wish to use another payment solution this would be managed through an extension of that client implementation.

### Requirement scenarios

1. The Order Processing application supports integration with the Picasso Payment Solution as part of a standard client implementation.
2. The Order Processing application would use the fulfilment and refund options with the payment integration.
3. The Order Processing application cannot support a client implementation which has their own payment solution available without an extension being developed, as each integration is different. Due to the nature of the security of payment gateways etc. each payment gateway would need to be assessed for the risks and security impact.
4. This has no impact on the reporting tool, as payment information is not sent to the business intelligence tool from the Order Processing application.
5. The integration with the Picasso Payment Solution will be monitored using the standard system health check and alert requirements.

## ERP integration

### Requirement statement

The Order Processing application will be integrated with the Optimus ERP.

### Requirement scenarios

1. At the appropriate workflow step order (inc. product identifier) is sent to the ERP so that the ERP can create the order in the system.
2. At the appropriate workflow steps information is retrieved or request sent to the ERP to enable order status be checked for an order.
3. The ERP will also support receiving a cancellation request for an order.
4. The Order Processing application assumes that the ERP has all of the necessary products configured (of all types e.g. physical products, price plans, services etc.).
5. There is a standard format for the integration.
6. If the client requires integration with a client specific Product System this would be managed through an extension.

## Comms Module

### Requirement statement

The Order Processing application will send emails to the customer, or as necessary to administrators, through integration with an communication system e.g. the Optimus Comms Module.

### Requirement scenarios

1. As per the configuration set up for the System Administrator for each workflow step the necessary emails (and email data) is sent to the Comms Module for an email to be sent to the customer.
2. There is a standard format for the integration.
3. If the client requires integration with a client specific Product System this would be managed through an extension.

# Requirement Specifications: B2B

The management of B2B sales through the order processing system will not be supported in phase 1.

# Requirement Specifications: Other services

There is no support for the management of other services through the order processing system in phase 1.

# Requirement Specification: Generic

## Generic requirements

### Requirement statement

The Order Processing application will adhere to the generic Optimus requirements (see separate HLR).

### Requirement scenarios

1. The following generic requirements are supported:
   1. Multi-region
   2. Multi-language
   3. Multi-currency
   4. Omni-channel
   5. Multi-tenanted
   6. Multi-site
   7. Audit logging
   8. Sales tax
2. The system will be monitored, as per:
   1. System health check
   2. System alerts
3. The system will support reporting, as per:
   1. Strategic reporting
4. There following generic requirements are not supported:
   1. Multi-device
   2. Operational information
5. A System Administrator can configure for an email to be automatically sent to the customer (or specific distribution list) via the Comms Module once this workflow step is completed.

# Glossary

| **Term** | **Description** |
| --- | --- |
| Back order |  |
| Channel |  |
| Client |  |
| Comms Module |  |
| Customer |  |
| eCommerce core |  |
| ePos |  |
| ERP |  |
| Fraud check |  |
| Fraud Management |  |
| Fulfilment (payment) |  |
| Logistics status |  |
| Middleware |  |
| Operational reporting |  |
| Order Processing Administrator |  |
| Order Workflow Step |  |
| payment gateway |  |
| Picasso Payment Solution |  |
| Pick and pack |  |
| Pre-auth |  |
| Product |  |
| Product Catalogue application |  |
| Region |  |
| Return |  |
| Shipment |  |
| Stock Orchestration Module |  |
| Strategic reporting |  |
| System Administrator |  |
| Voucher |  |

# Data Confidentiality Statement for this document

## Confidentiality Policy

This data confidentiality policy is intended to:

* Protect sensitive company data
* Support the communication of company data on a ‘need to know basis’

This policy is applicable to all information within and relating to the **High Level Requirement: Optimus Order Processing**. Failure to comply with the policy may place the EXPANSYS group companies and subsidiaries, suppliers and clients at serious risk and/or result in financial loss. Failure to comply with the policy when handling data may be considered a disciplinary offence.

This Policy applies equally to EXPANSYS, PJ Media Limited, PJ Interactive and Data Select.

The confidentiality level of this document has been determined as Category 1 – Critical: This data, if lost or published outside of its intended audience will result in significant financial loss or reduction in the company’s ability to execute its mission. The audience for this document is limited to those listed below.

The business owner for the High Level Requirements: Optimus Order Processing is Piotr Ignaczak; all queries relating to the confidentiality should be directed to this person.

## Intended audience

[If this is restricted 3rd party and unrestricted internally, then modify this section to be restricted 3rd party audience only.]

| **Company** | **Role** | **Name** |
| --- | --- | --- |
| **[Company]** | **[role of document owner]** | **[name]** |
| PJ Media Limited | Business Analyst | [name] |
| [company] | [role] | [name] |

## Intended audience audit

| **Version #** | **Change Description** | **Date** | **Author** |
| --- | --- | --- | --- |
| 0.1 | Created intended audience | dd/mm/yy | [name] |