SAP Commerce 2005 Connector R4.0 - Technical Implementation Guide

Reader's guide

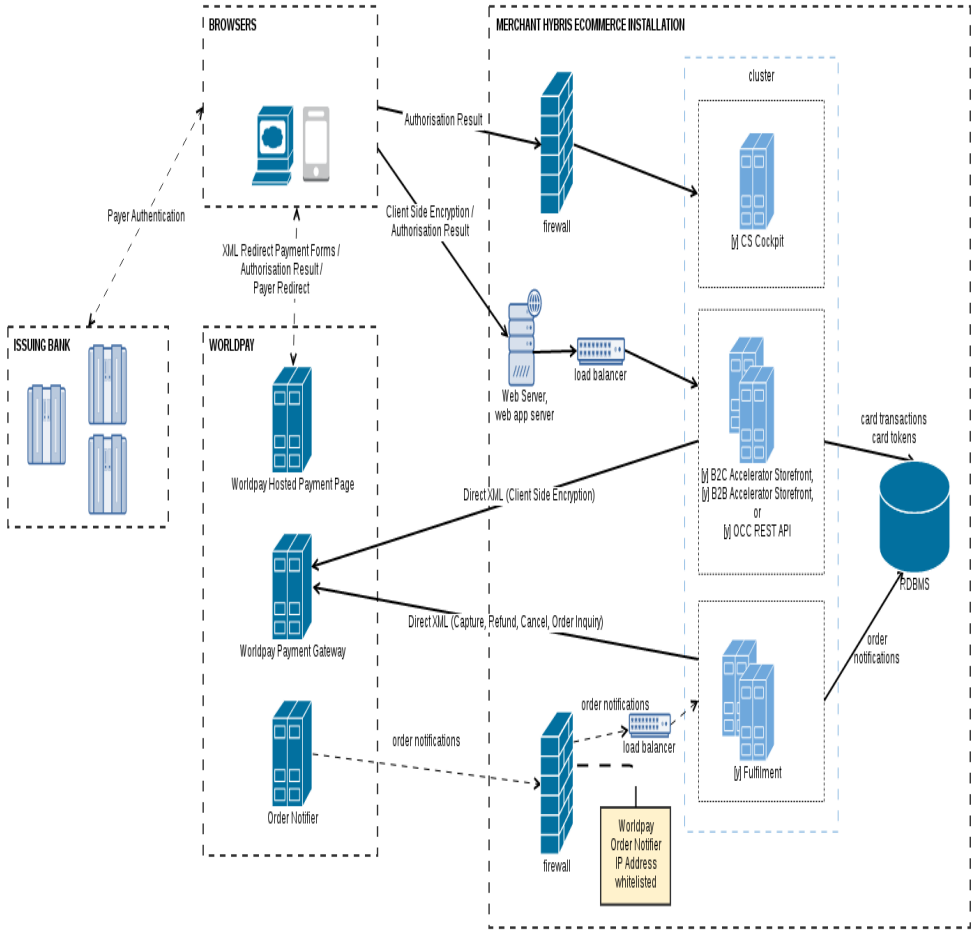
This technical implementation guide will cover the use of Worldpay as a payment provider mainly in a B2C context. Some sections will apply to both B2C and B2B that only apply to either B2C or B2B will be marked as ***B2C only*** or ***B2B only***

Technical Implementation Guide

All Worldpay Payment Commands are executed asynchronously and therefore will respond initially in a *pending* state. Once the transaction has completed, Worldpay pushes notifications via the Order Notifier back to a configured endpoint on a Hybris Commerce Suite node. The endpoint URL is configured in the merchant account in the Worldpay Merchant Account Interface Tool. The endpoint (exposed by installing the **worldpayn otifications** extension) serialises these notifications as **WorldpayOrderModification** records into the Hybris Commerce Suite database.

High-Level Architecture

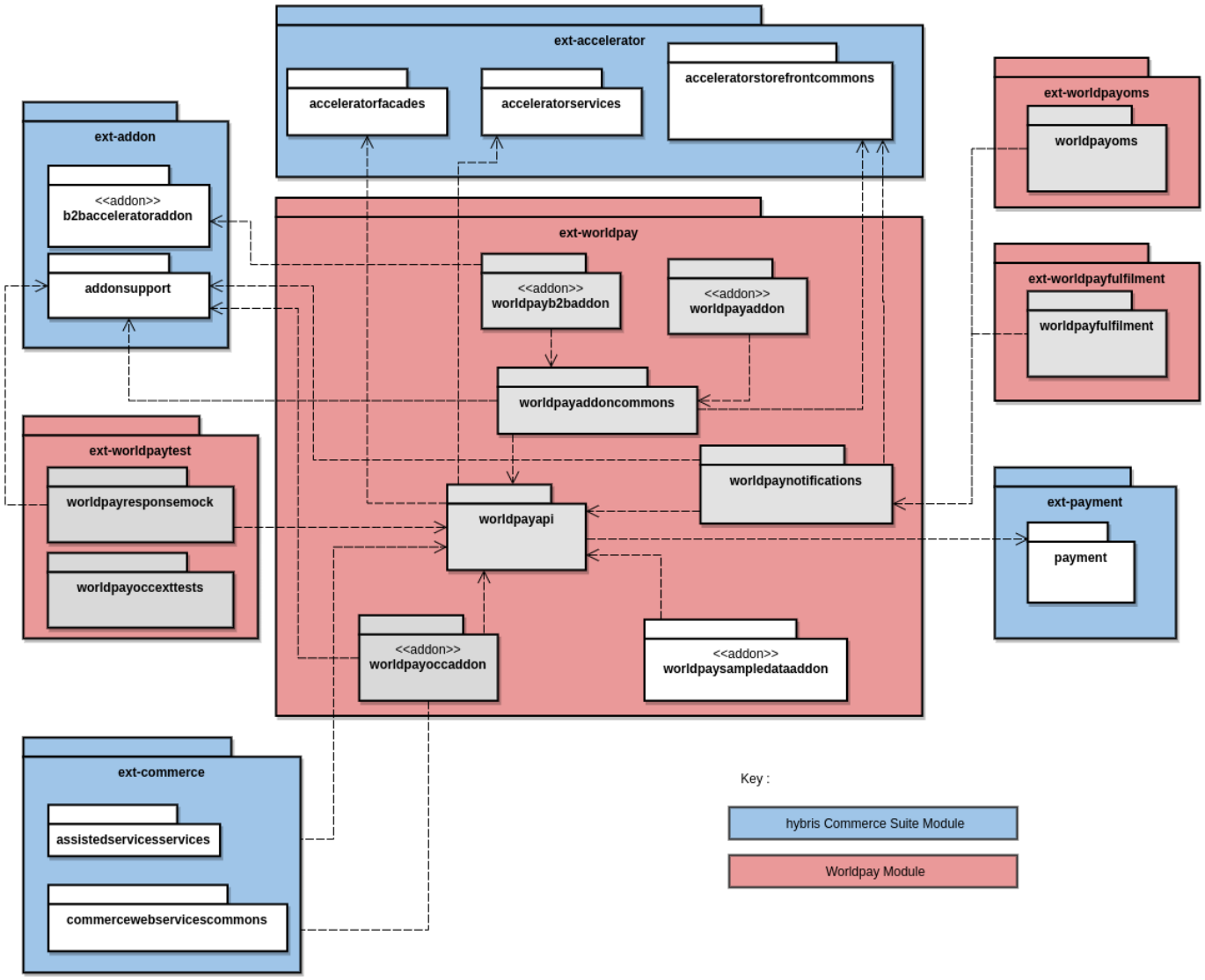
The following diagram outlines at a high level, the system components (Worldpay, Issuing Bank and Hybris) together with the high-level interactions that fulfil the functionality delivered by the Worldly AddOn.



Hybris Commerce Suite Extensions

The Worldpay Hybris AddOn is shipped in 5 separate modules **ext-worldpay, ext-worldpayfulfilment, ext-worldpayoms, ext-worldpaytest**.

The Worldpay AddOn extends the Hybris Commerce Suite providing integration with Worldpay Hosted Order Page, Payment API, Notification, Fraud, CSE and tokenisation functionalities.



Connector

worldpayapi

Worldpay API Gateway Plugin point for the Hybris Commerce Suite Payment Service.

Related Documentation

- Payment Extension

Front end

worldpayaddon

AddOn for the Hybris B2C Accelerator. This adds a new Checkout Flow that is optimised for Worldpay HOP (Hosted Order Page), CSE/Direct both supporting tokenisation functionality.

Related Documentation

- Configurable Checkout
- AddOn Concept

worldpayb2baddon

AddOn for the Hybris B2B Accelerator. This adds a new Checkout Flow that is optimised for Worldpay CSE/Direct supporting tokenisation functionality.

Related Documentation

- Configurable Checkout
- AddOn Concept
- B2B Checkout and Order Process

worldpayaddoncommons

Commons extension for B2C and B2B AddOn's.

worldpayextocc

Extension for the Hybris OCC REST API. This adds payment details and place order in a B2C context. Support CSE and 3D secure.

Related Documentation

- OCC Architecture Overview
- OCC extension

worldpaysampledataaddon

AddOn that adds sample data on the electronics and apparel sites. Contains several APM configurations, adds some countries, currencies and delivery zones. Also adds CMS components to the payment and billing page for the electronics and apparel content catalogs enabling the different APMs.

The data provided here could serve as an example of how to import different APMs and how to configure the different components in the payment and billing page.

worldpayfulfilment

Adds Worldpay related functionalities and customisations to payment integrations for the Hybris Accelerator Order Fulfilment Process.

Related Documentation

- yacceleratorfulfilmentprocess Extension

worldpayoms

Adds Worldpay related functionalities and customisations to payment integrations for the Hybris Order Management System.

Related Documentation

- Order Management for SAP Hybris Commerce

worldpayaddonbackoffice

Extension that includes configuration files so items created in the AddOn are organised and visible in the backoffice Hybris application.

Related Documentation

- Backoffice Framework

Notification

worldpaynotifications

Extension providing an HTTP endpoint to receive and process Worldpay Order Notification messages.

Testing

worldpayoccexttests

Extension containing Spock test that verifies the endpoints implemented at the **worldpayextocc.** See our OCC Documentation for usage and details.

worldpayresponsemock

The connector contains an extension that offers two functionalities:

Direct/redirect responses mock

Acting as the Gateway itself and replying to Direct and Redirect requests. The mock replies with pre-fabricated happy-flow responses to authorize and capture requests.

To enable the mock, set the following properties:

worldpayapi/project.properties

```
# Valid values for environment are MOCK, TEST and PRODUCTION
worldpay.config.environment=MOCK
worldpay.config.endpoint.MOCK=http://<YOUR-SITE>:9001
/worldpayresponsemock/mock
```

You also need the add the property -Djavax.xml.accessExternalDTD=all to the property tomcat.generaloptions in you local.properties

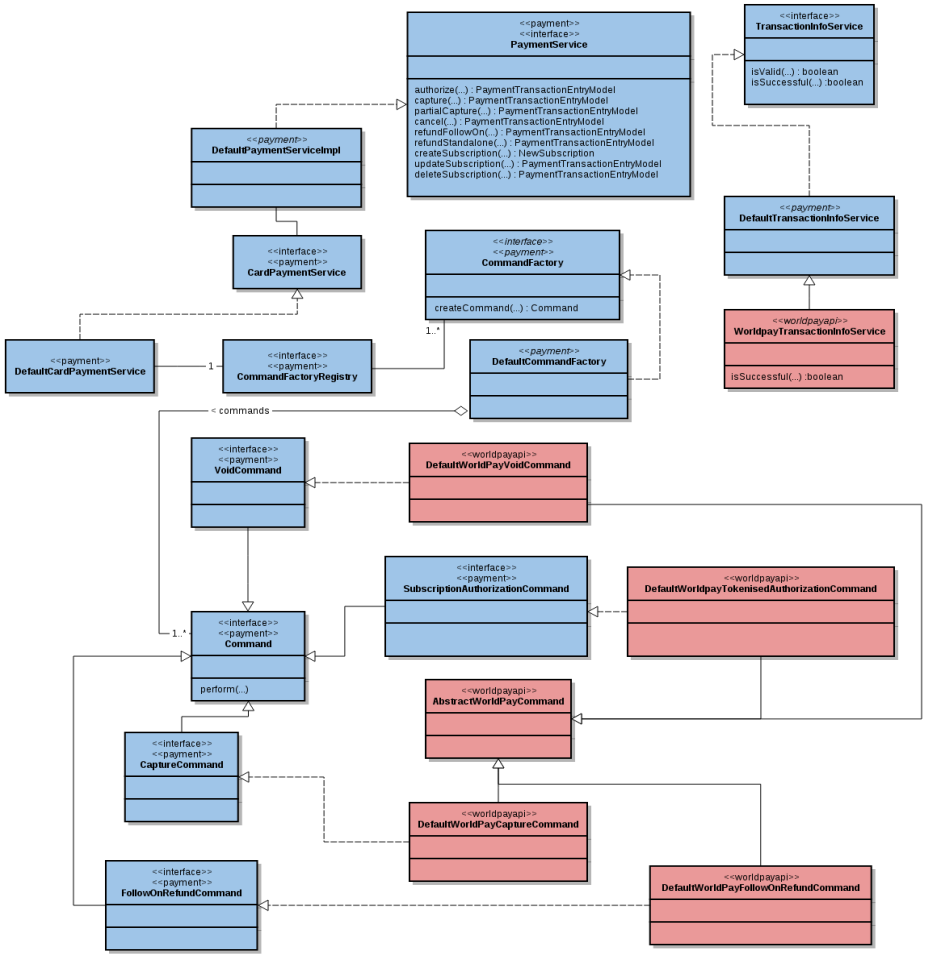
Order notification mock

The Worldpay Order Notification System requires the Hybris Commerce Suite to expose an internet-facing Webservice endpoint. To facilitate development and test automation, the AddOn ships a test harness to mockup notification system responses in development environments. This is an optional installed Hybris extension with a simple Web front-end that can be used to simulate order notifications updates from Worldpay. Write in the Order Code (requestId, found in the backoffice or in the logs) and send the response you wish generated.

AddOn Integration with Hybris Payment Module

The Worldpay extension plugs directly into the Hybris Payment Module by providing Worldpay connectivity to authorization, capture, refund, void and subscription commands. The Worldpay implementations of these commands then use the Core Connector Library to communicate with the Worldpay Service Gateway.

The following class diagram shows the points in which Worldpay plugs into the Hybris Payment Service.

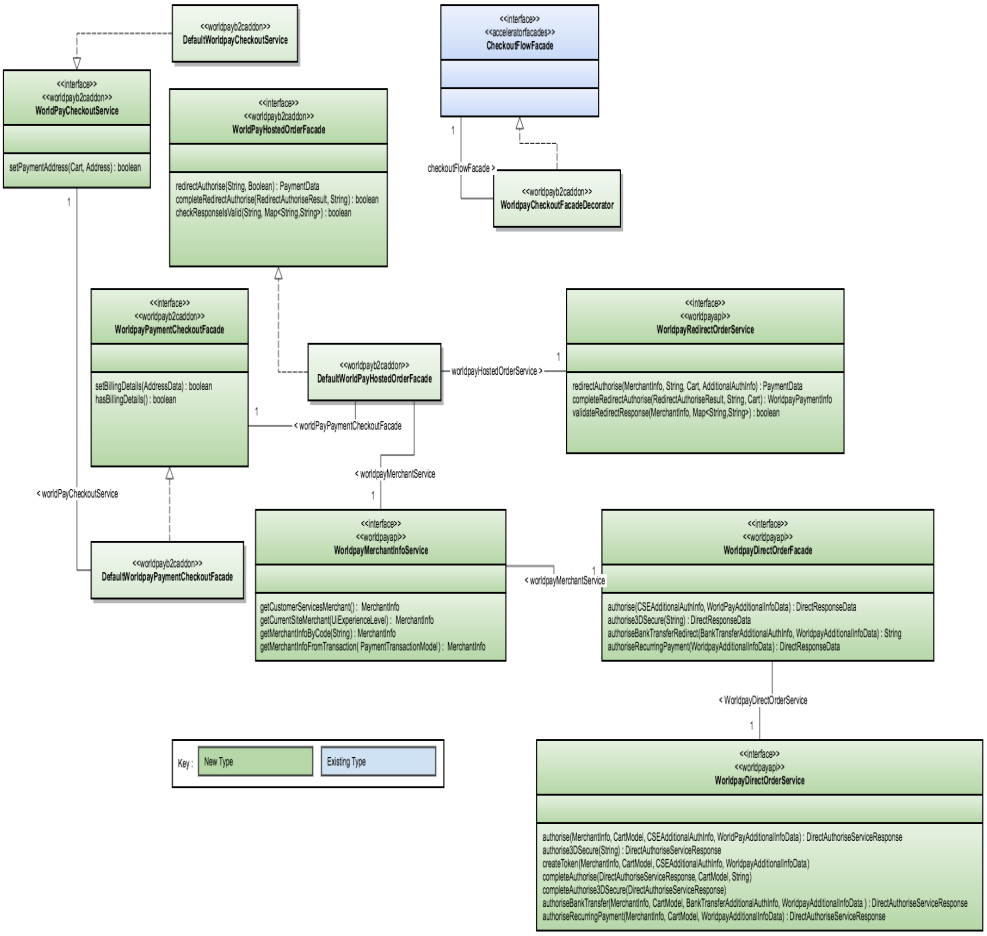


It is important to note, with the Worldpay integration, confirmation of the result of executing the command comes asynchronously via Worldpay posting to the Order Notification AddOn. Therefore Commands will return Payment Transaction Entries with a pending status.

New Accelerator Checkout Flow

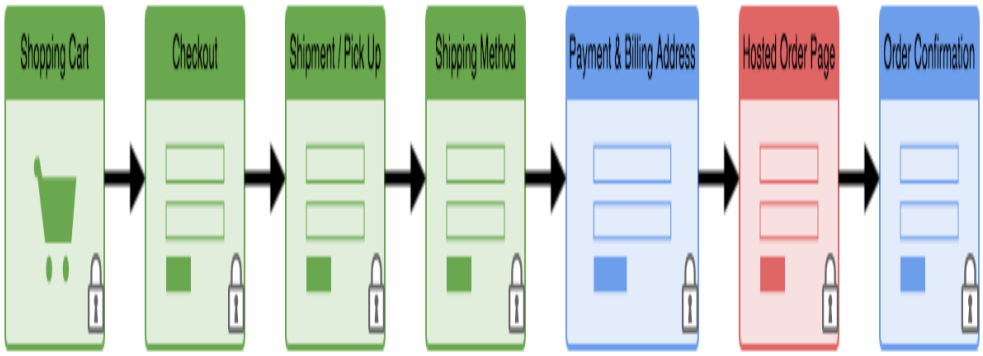
Facades & Services - *B2C only*

The Accelerator Service Layer and Facades have been extended to support functionality including XML Redirect Hosted Order Payment, Pay As Order and an XML Direct using Client-side encryption.



HOP - Redirect Flow - *B2C only*

The following diagram depicts the modifications made by the Worldpay B2C Accelerator AddOn which adds a Worldpay compatible Checkout Flow to replace the standard B2C Accelerator flow.



Payment & Billing Address Page - *B2C only*

This page has been customised to:

- Display a list of available payment methods (such as Cards and Alternative Payment Methods) which are CMS-driven.
- Display Terms & Conditions checkbox which is a required field in order to proceed the Hosted Order Page.
- JavaScript library added to handle the User Experience.

Order Summary Page - *B2C only*

This page has been removed from the vanilla Accelerator HOP flow. This is because the authorisation happens on the Worldpay Hosted Order Page and there is no need to ask the User to place the order. The order is placed at the end of the HOP page callback.

Order Confirmation Page - *B2C only*

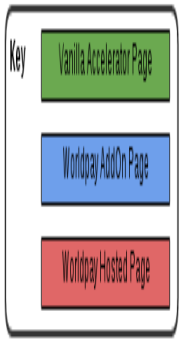
Small customisation to hide Payment Details in the event of the Notification not yet being received. Once the application has received the notification, it will show either the payment details about the credit/debit card used (obfuscated) or the Alternative Payment Method used to place the order.

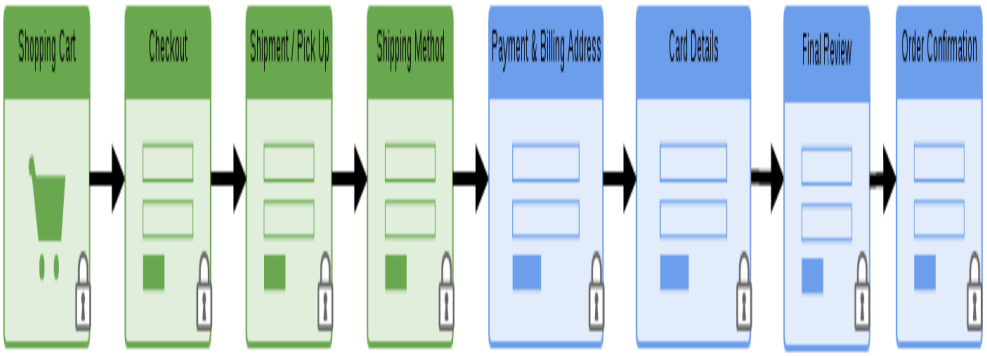
Tokenisation - *B2C only*

When the customer chooses to save their card details in the Payment and Billing page, the request sent to Worldpay contains an instruction to tokenise the card. The corresponding authorisation order modification message will contain the token information.

Client-Side Encryption Flow

The following diagram shows how the Client-Side Encryption (CSE) payment flow has been implemented in the AddOn. This flow is only available for credit/debit card payments.





The only difference from the redirect-flow is that the user is no longer taken to the Hosted Order Page for filling in their card details. Card details are instead encrypted on the customer's browser and the encrypted payload is submitted to the Merchants Web storefront by a form post on the Card Details page. This encrypted data is then passed on to Worldpay using Direct XML integration.

Tokenisation

When the customer chooses to save their card details in the Card Details page, the direct request sent to Worldpay contains an instruction to tokenise the card. The direct response will contain the token information. The AddOn will store this information in the payment info associated with the order.

The response from Worldpay may contain a tokenEvent with the value "NEW" or "MATCH". NEW means that is a new card tokenised and stored by Worldpay, whereas MATCH means that the card has already been tokenised and the customer is not using an already saved card. In this case, the AddOn will look for an already tokenised and saved card associated to the customer and will attach the found one to the order and payment transaction. If there is no such payment info associated to the customer, a new one with the token information will be created.

Notifications

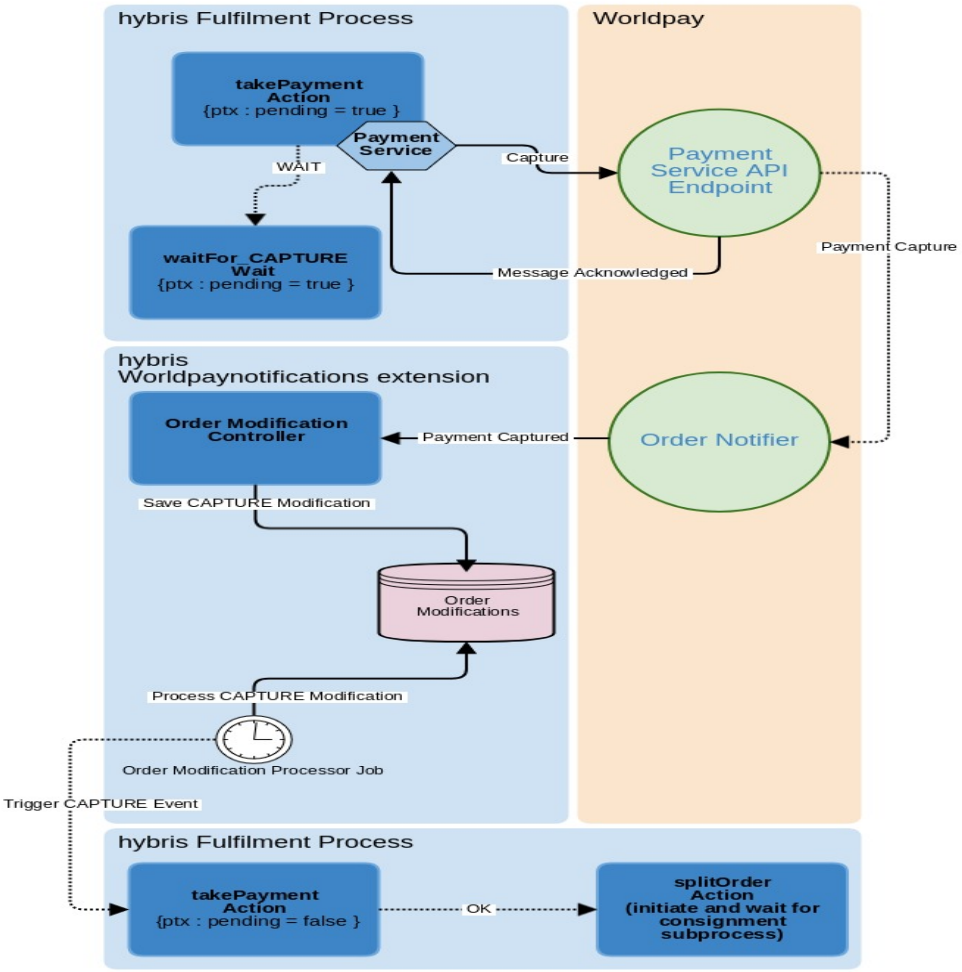
All Worldpay Payment Commands are executed asynchronously and therefore will respond initially in a *pending* state. Once the transaction has completed, Worldpay pushes notifications via the Order Notifier back to a configured endpoint on a Hybris Commerce Suite node. The endpoint URL is configured in the merchant account in the Worldpay Merchant Account Interface Tool. The Endpoint (exposed by installing the **worldpayn otifications** extension) serialises these notifications as **WorldpayOrderMofication** records into the Hybris Commerce Suite database.

Tokenisation

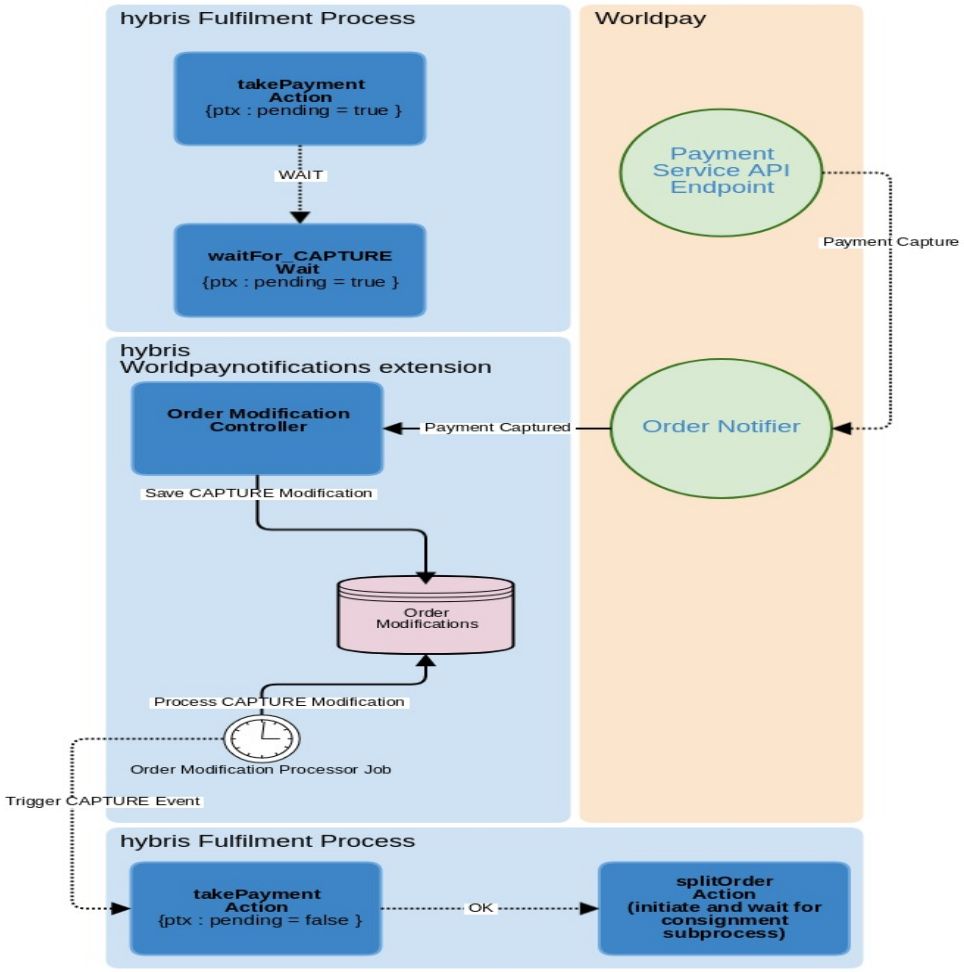
When the application receives the notification with the token information in the Redirect Flow, the AddOn will store this information in the payment info associated to the order.

The response from Worldpay may contain a tokenEvent with the value "NEW" or "MATCH". NEW means that is a new card tokenised and stored by Worldpay, whereas MATCH means that the card has already been tokenised and the customer is not using an already saved card. In this case, the AddOn will look for an already tokenised and saved card associated to the customer and will attach the found one to the order and payment transaction. If there is no such payment info associated to the customer, a new one with the token information will be created.

To illustrate the flow of a typical payment capture request for a **credit/debit card payment** use-case from Hybris Commerce Suite to Worldpay and then back into Hybris Commerce Suite, the following diagram is shown:

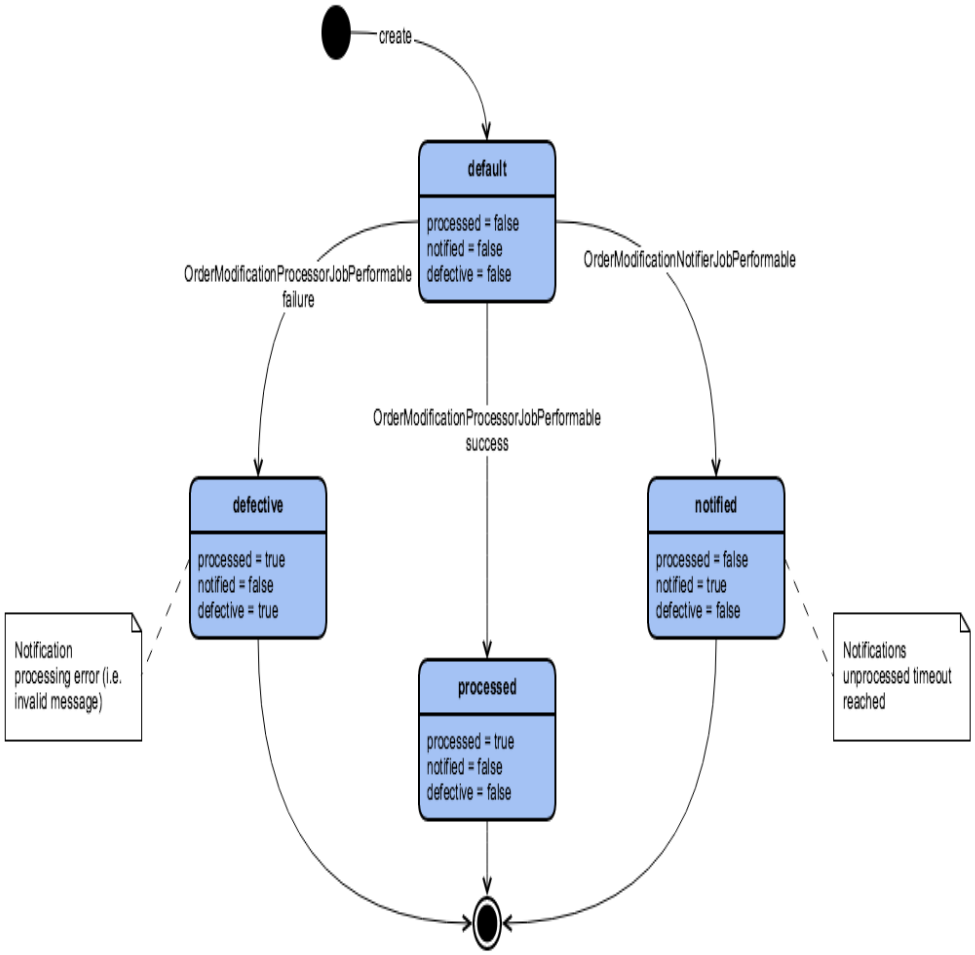


To illustrate the flow of a typical payment capture request for an **Alternative Payment Method** (**APM) payment** use-case from Hybris to Worldpay and then back into Hybris, the following diagram is shown:



Notification States

The following state diagram describes the states, transitions and possible attribute values of a **WorldpayOrderModification** record in the Hybris Commerce Suite database.



Notification Cleanup

WorldpayOrderModification's can build up over time, therefore two types of cleanup Cron Job's have been provided with the AddOn.

- Cleanup of processed notifications
  - A scheduled Cron Job will retrieve the Modification records which have been processed after a certain amount of time (configurable) and remove them from the system.
- Cleanup of unprocessed notifications
  - A scheduled Cron Job will retrieve Modification records which have been sitting unprocessed for a certain amount of time (configurable) and raise a Ticket for each of them.

Order Fulfilment Process

To help accelerator development of best practice payment fulfilment workflows, reference *Order Fulfilment Processes* have been supplied with the AddOn. Depending on whether you use Hybris Order Management System or the Accelerator Fulfilment Process you can add the **worldpayoms** or the **worldpayfulfilment** extension into your installation (see Installation and Usage).

When an order is placed in the Hybris Commerce Suite, it is sent to the process engine for execution by the order fulfilment process. There, the order is processed by a number of actions each having a different purpose. In WorldpayCheckAuthorizeOrderPaymentAction the amount from PaymentTransactionEntries - of type AUTHORIZED - will be compared to order total. If they don't match the order status will change to CHECKED_INVALID, and an OrderHistoryEntry will be added. This could happen if the cart has been modified during payment. For error handling, the PaymentTransactions can be found in Backoffice on the order Administration tab.

⚠ When comparing order total and authorised amount the difference is validated against a tolerance. The default tolerance is 0.01, but the tolerance level is configurable by property

```
worldpayapi.authoriseamount.validation.tolerance=0.01
```

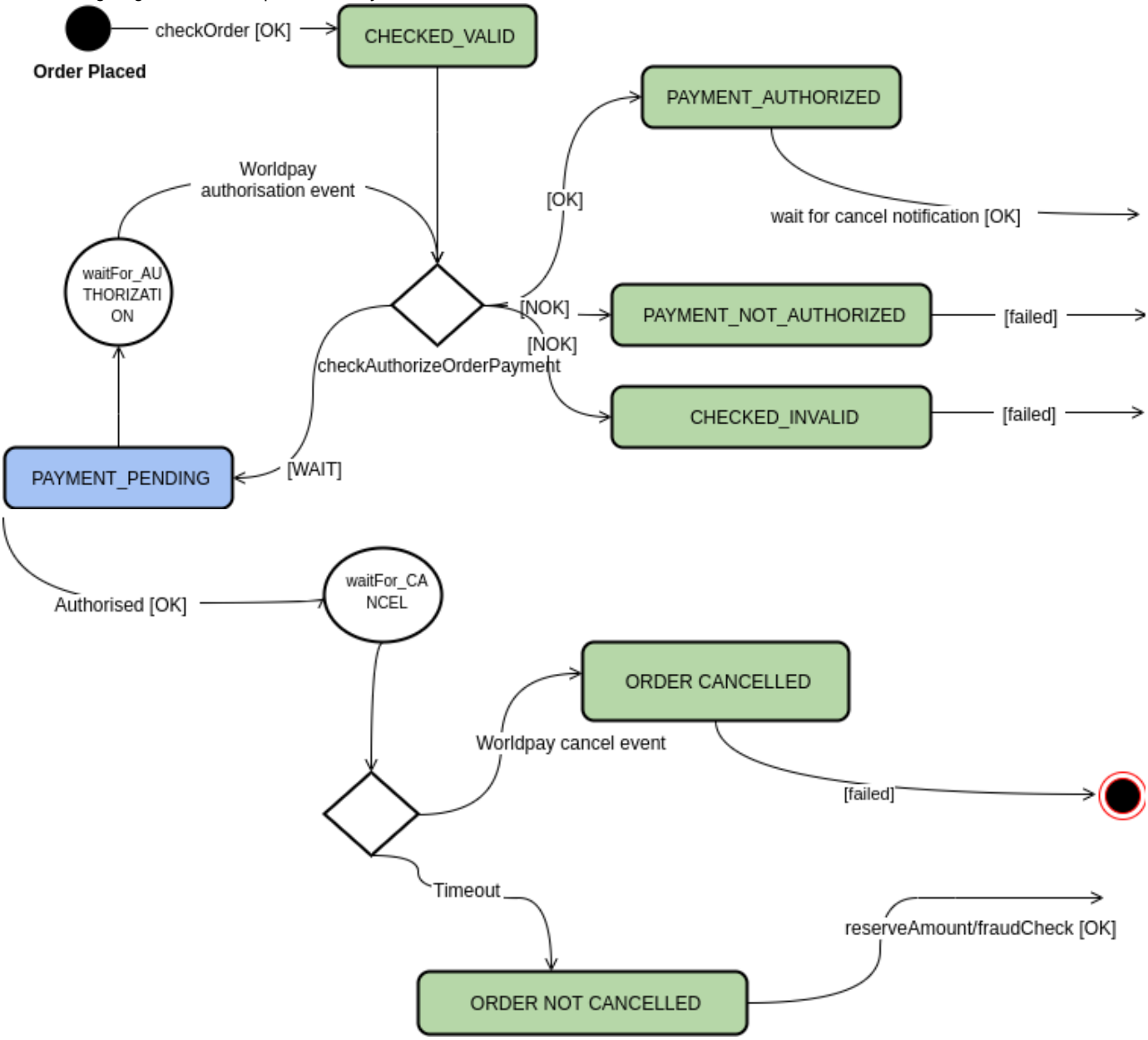
When all the actions have been executed, the order reaches its final state (be it error or success).

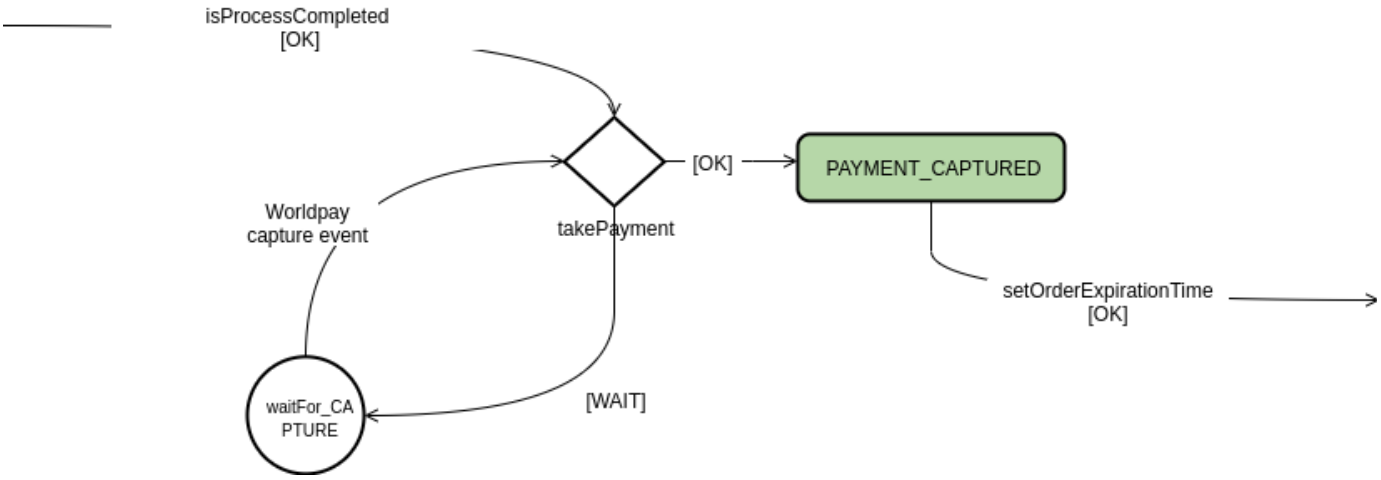
Given the asynchronous nature of Worldpay Order Modification messages, a number of WAIT states have been added to the reference process. Since every payment transaction entry is created in a pending state (exception for APM's), the WAIT will pause the fulfilment process until the confirmation of the said payment transaction has been confirmed (or denied). There is one WAIT for every Order Modification message we handle in the AddOn (AUTHORIZATION, CANCEL and CAPTURE). When the AUTHORIZATION message has been processed the *Order Fulfilment Process* waits for a possible cancel notification, but once it has reach the timeout for waiting for a cancel notification it continues until the **takePayment** step.

When an Order Modification message is received from Worldpay, the application persists the Modification message in the database and returns an acknowledgement (OK) response to the Order Notifier. A Cron Job (Order Modification Processor Job) is responsible to go through the unprocessed Order Modifications and process them. Different type of messages require different processing but for all of them, once processed, an event is triggered which the WAIT is waiting for. Once the WAIT receives the event, the Order Fulfilment process continues.

In the special case of APM, the order process will wait for the CAPTURE modification message, when it reaches the application, the transaction entry is created in a non-pending state. The order process will not issue a CAPTURE command to Worldpay when the payment was made through an APM except Klarna APM, which actually does a CAPTURE command.

The following diagrams show the parts of the Hybris Order Process where the WAIT states have been added:





Every order for which the authorisation notification has not been received from Worldpay will have the PAYMENT_PENDING status set and will stay in the waitFor_AUTHORIZATION wait state until such notification has been received. Depending on the outcome of the notification, the order processing will continue (authorisation successful) or will fail (authorisation not successful).

Return process

There is a customised return-process available if the chosen AddOn installation contains the OMS functionality. The modifications applied to this return-process adds a new waiting step for the REFUND notification.

worldpayoms/process/return-process.xml

```
     <wait id="waitForConfirmOrCancelReturnAction" prependProcessCode="
true" then="failed">
        <case event="ConfirmOrCancelRefundEvent">
            <choice id="cancelReturn" then="cancelReturnAction"/>
            <choice id="approveReturn" then="approveReturnAction"/>
        </case>
    </wait>

    <action id="cancelReturnAction" bean="cancelReturnAction">
        <transition name="OK" to="success"/>
    </action>

    <action id="approveReturnAction" bean="approveReturnAction">
        <transition name="OK" to="printReturnLabelAction"/>
    </action>

    <action id="printReturnLabelAction" bean="printReturnLabelAction">
        <transition name="OK" to="printPackingLabelAction"/>
    </action>

    <action id="printPackingLabelAction" bean="printPackingLabelAction">
        <transition name="OK" to="waitForGoodsAction"/>
    </action>

    <wait id="waitForGoodsAction" prependProcessCode="true" then="
failed">
```

```
        <case event="ApproveOrCancelGoodsEvent">
            <choice id="cancelReturn" then="cancelReturnAction"/>
            <choice id="acceptGoods" then="acceptGoodsAction"/>
        </case>
    </wait>

    <action id="acceptGoodsAction" bean="acceptGoodsAction">
        <transition name="OK" to="captureRefundAction"/>
    </action>

    <action id="captureRefundAction" bean="captureRefundAction">
        <transition name="OK" to="waitFor_REFUNDED"/>
        <transition name="NOK" to="waitForFailCaptureAction"/>
    </action>

    <!-- New WAIT to be kicked off when the transaction is Refunded by
worldpay -->
    <wait id="waitFor_REFUNDED" then="successCaptureAction"
prependProcessCode="false">
        <event>${process.code}_REFUND_FOLLOW_ON</event>
    </wait>

    <wait id="waitForFailCaptureAction" prependProcessCode="true" then="
failed">
        <case event="FailCaptureActionEvent">
            <choice id="bypassCapture" then="taxReverseAction"/>
            <choice id="cancelReturn" then="cancelReturnAction"/>
        </case>
    </wait>
```

The action captureRefundAction is also customised:

worldpayoms-spring.xml

```
    <alias name="worldpayCaptureRefundAction" alias="
captureRefundAction"/>
    <bean name="worldpayCaptureRefundAction" class="com.worldpay.
worldpayoms.fulfilmentprocess.actions.order.
WorldpayCaptureRefundAction" parent="abstractAction">
        <property name="paymentService" ref="paymentService"/>
        <property name="refundAmountCalculationService" ref="
refundAmountCalculationService"/>
    </bean>
```

Order Cancellation

The Worldpay AddOn enables the Call Centre Agents to cancel orders. Cancelling an order in Hybris sends a CANCEL request to Worldpay to reverse ringfenced funds on the customers payment method.

If the order was paid through an APM, or the payment method has not yet been communicated by the Worldpay Notification System, the order cannot be cancelled with the out of the box handling of the AddOn. If the Hybris installation is customised to handle multiple payments for the same order and one of these is an APM, the order will not be cancellable. This functionality can of course be changed on project once business rules have been realised.

In case of fraud review, the "Reject" action in the Backoffice will place the order in a WaitForCleanUp state. By default Hybris cancels the order through the cleanUpFraudOrderCronJob triggering the "VoidCommand" action, which is customised to send a CANCEL request to Worldpay.

Risk Guardian

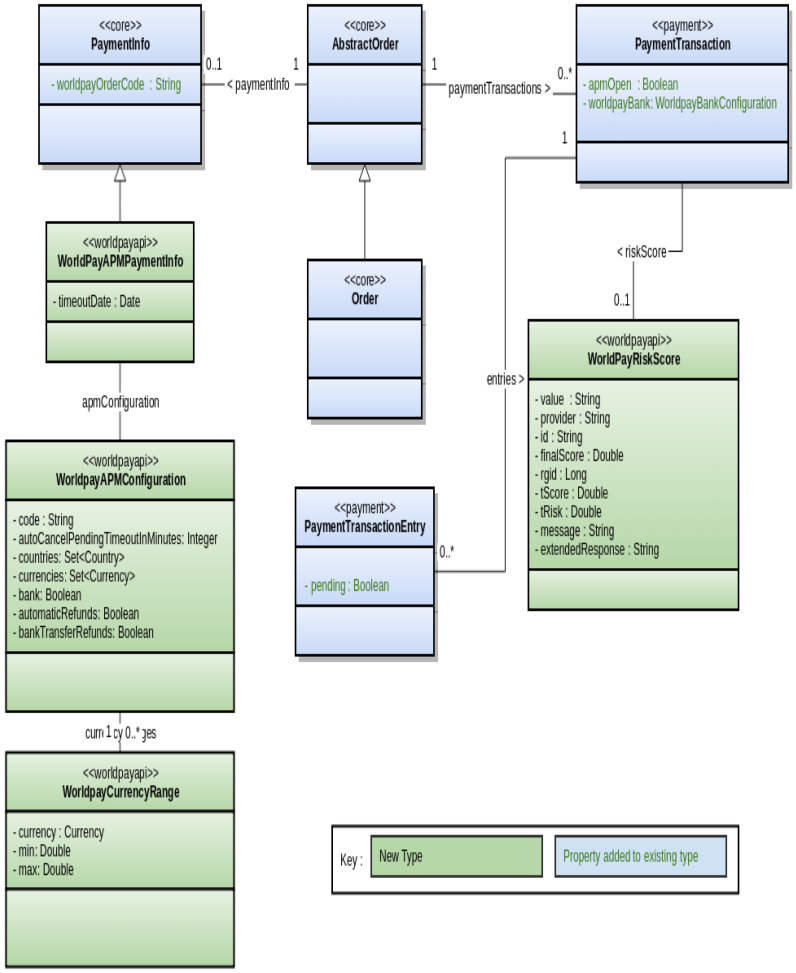
Risk Guardian is an advanced fraud detection tool that can be activated on a **Production** Worldpay Merchant Account. Extra fraud score information is returned with the Authorisation Response which is then captured against the Payment Transaction.

Out of the box the information is captured and displayed (in Customer Service tools) and orders with fraud scores greater than a configured threshold (**project.properties**) are held for review.

Extended Types

The Worldpay AddOn extends the Hybris Commerce Suite data model in a few area's (predominantly payment).

The main customisations are to introduce the concept of Pending Payment transactions, Fraud Report information and to support a more generic Payment type accommodating Worldpay's support for 200+ Alternative Payment Methods.



Update Worldpay's DTD

From time to time Worldpay releases a new version of their DTD, the definition of their XMLs.

Worldpay's DTD: http://dtd.worldpay.com/v1/

What's a DTD: https://en.wikipedia.org/wiki/Document_type_definition

Under `y-ext/ext-worldpay/worldpayapi/resources/dtd` you can find 2 files, the original from Worldpay, so we can track their changes: `paymentService_v1_wp_original.dtd`
and a java compatible one with the name `paymentService_v1.dtd`. This file is used in the buildcallbacks in `y-ext/ext-worldpay/worldpayapi/buildcallbacks.xml` to generate the classes to handle the XMLs from its definition.

If there is a change in the DTD, a new XSD file needs to be generated. The XSD file is used to validate the object representing the XML request to Worldpay.

This can be achieved by running `ant -f schemagen-build.xml`. This task will generate a file named "paymentService_v1.xsd" under `resources/schema`.

Reference: https://en.wikipedia.org/wiki/XML_Schema_(W3C)

Supported Environments

Third-party compatibility

Limitations

The Worldpay Payment Service API has different field length restrictions to the Hybris Accelerator / Commerce Suite. Email for example has a maximum length of 30 characters but Hybris accepts 255. The AddOn does not add any extra logic to check the length of these fields and this is expected to be customised on a project-by-project basis.

Languages

The provided AddOn only provides localization in English. The translations for other languages can be added by the integrator in the corresponding properties files on a project basis.