Google Checkout Integration: High Level Architecture

## High Level Goals

* Offer Google Checkout as an option for customer payment during the checkout process on one of the e-commerce websites.
* Integration with Google Checkout without the need for enhancement of the SAP system.
* Utilize infrastructure from the Amazon/Mercent e-commerce integration where possible, to for a consistent checkout pipeline and to keep development costs under control.

## Major Components

**Google Checkout:** A payment processing service offered by Google. Google provides the user experience for a buyer checking out, collecting payment information on our behalf. Interaction with Google takes place asynchronously via REST-based web services. Communication is bi-directional with Google both receiving and sending order information.

**SAP Staging Database:** Conceptualized and created for the Amazon/Mercent integration project, this database serves as a queue for order-related transactions and updates. SAP will monitor this database and consume any transactions posted for it, integrating them into the standard order processing operations. In addition, SAP will post updates for orders received via this staging database, which will trigger actions to be requested from Google Checkout.

**Payment Services Framework:** A framework serving the role of listener and translator, allowing SAP and Google Checkout to communicate without either needing to understand the other’s interface. Further, the goal is to provide a lightweight wrapper with a neutral interface, allowing other third party payment processors to be added with minimal new development needed. The framework consists of a client library, scheduled job, and REST-based web services (both internal and public-facing).

## Order Flow

1. **Placement**
   1. On one of the e-commerce websites, a customer places and order and clicks the button for the Google Checkout payment method.
   2. The website calls the Payment Services Framework client library, which invokes an internal-facing REST service. The internal service will submit a checkout request to Google.
   3. When the Checkout Request is received, Google validates the information and synchronously responds with a url for the user to complete the checkout process. This url is returned by the internal REST service to its caller.
   4. The e-commerce site uses the Google-supplied url that it receives from the Payment Services client library call and redirects the customer using a 302-type redirect.
   5. During the process of checking out the customer, the customer provides payment information to Google, which they will interpret and verify with the credit card companies. As a part of the checkout process, Google will invoke the external-facing REST service to calculate the taxes, shipping charges, and discounts to apply to the order.
2. **Acceptance**
   1. Once the payment flow has been completed, Google will invoke the public-facing REST service to notify [COMPANY] that a new order has been placed. The order notification is held in the staging database for inclusion in SAP’s order processing.
   2. After placement is complete, Google will perform a risk analysis on the customer and the order. When this is complete, Google will invoke our REST service and notify us of the result. The results are staged for inclusion in SAP’s order processing.
   3. The charge is authorized by the credit card company and Google. Google will invoke the external-facing REST service and notify [COMPANY] that the order is chargeable. This notification is for inclusion in SAP’s order processing. The authorization is valid for 168 hours (7 days) once this notification is sent.

1. **Charging**
   1. SAP stages a request to charge the order into the staging database, which is read by the polling job. The job will invoke an internal-facing REST service which issues the request to Google to charge the order. According to the merchant agreement, the customer may only be charged when the order is ready to ship. Partial charges may be placed.
   2. Google will acknowledge receiving the charge request by invoking the public-facing REST service. This notification will be staged for inclusion in SAP.
   3. Google will make an attempt to charge the order, as requested. Upon completion, Google will invoke two of our REST service endpoints. The first serves to notify us of the result of the attempt, the second notifies us of the amount charged. Both of these notifications will be staged for inclusion in SAP.
2. **Shipping**
   1. Once the order is shipped, SAP will stage an update to that effect. The polling job will receive this update and invoke an internal-facing REST service which passes the notification to Google.
   2. Google will acknowledge this update by invoking our REST service with an update to the order status. This will be staged for inclusion in SAP.
3. **Completion**
   1. Once the order has been closed, SAP will stage an update to that effect. The polling job will receive this update and invoke an internal facing REST service which requests that Google archive the order.

# High Level Integration Flow

