



Procurement

Using OCI connections

What is an OCI connection?

OCI is short for Open Catalogue Interface, which means that the Spend Cloud is able to connect to a webshop. People can use this connection to order products in your webshop without leaving the Spend Cloud. This is beneficial for the person placing orders, the organization they are ordering for, and for the supplier itself.

Why use OCI connections to improve processes?

It takes a lot of work to keep information updated when not using OCI connections, since you have to provide extensive administration. With every price change (temporary or permanent), suppliers have to contact customers and process data correctly. This makes the process prone to error. Corrections take up extra time and result in extra costs.

Processing with OCI connections means processing with data (products and agreements about price and volumes) that is always up to date and automatically available. The process is thus less prone to error. OCI connections are also not a Spend Cloud exclusive functionality. They are able to work for every customer willing to invest time and effort. A uniform way of working is within reach!

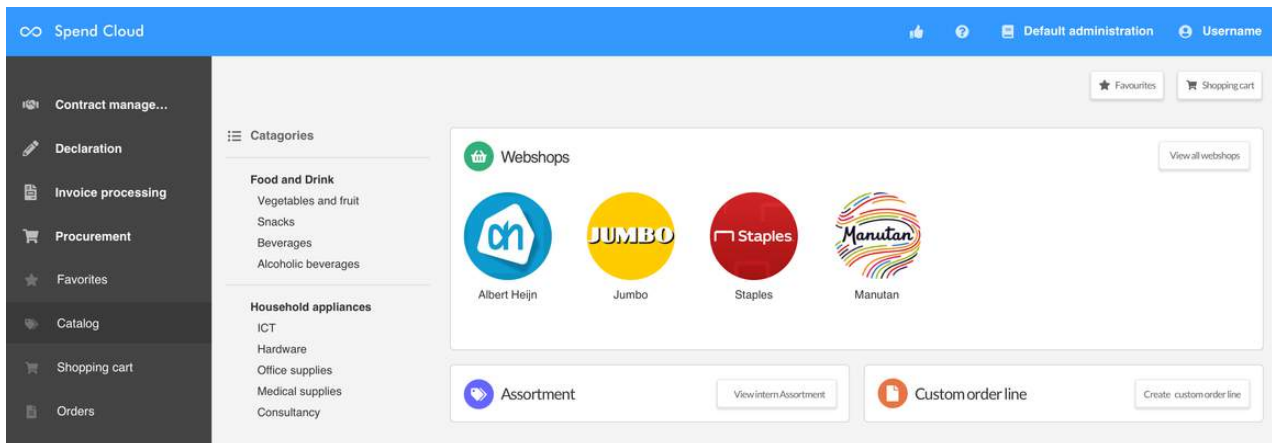
In short: OCI connections are the solution

- Customers process data that is up to date.
 - Temporary deals are immediately visible.
 - Temporary deals disappear immediately when they are no longer valid.
- Customers always pay valid prices.
- Processes are less prone to error.
- This saves time, effort, and money.
- You can implement uniform processes.
- This allows for more efficient processing of orders.



How do OCI connections work in the Spend Cloud?

A user logs in to the Spend Cloud and sees with which suppliers an OCI connection is set up. For suppliers with whom we don't have a connection, other orderings options are available such as predefined articles in their own catalog or a custom order line.

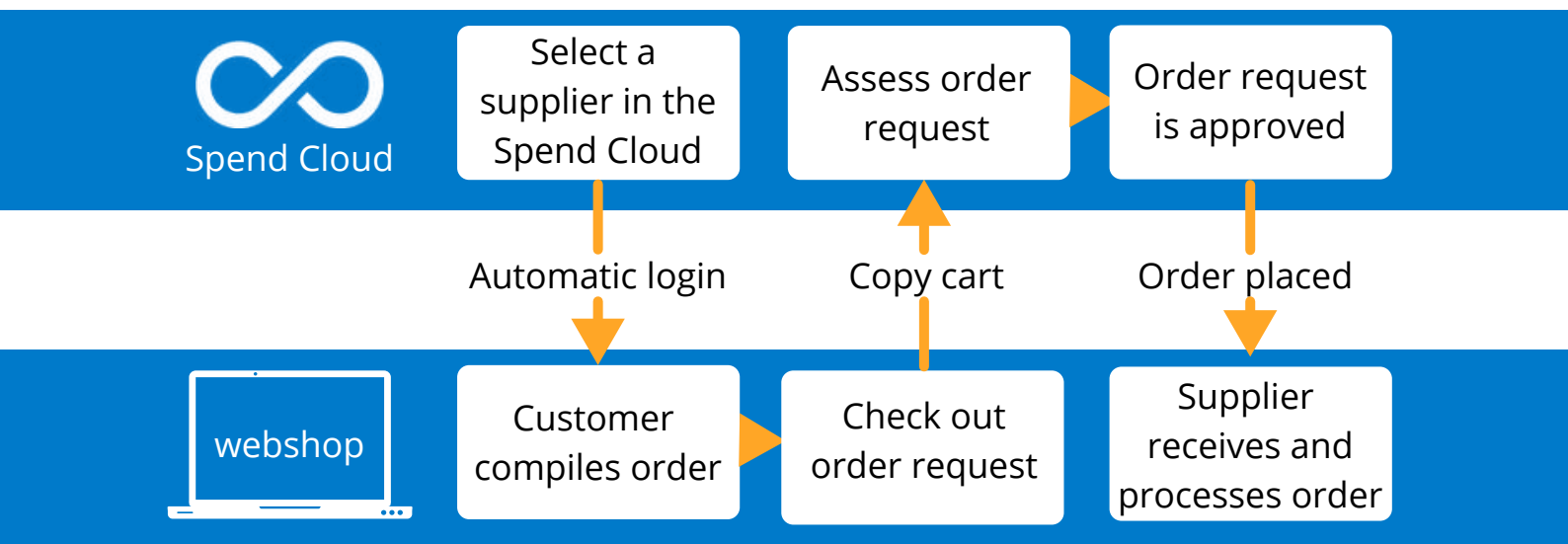


By clicking on the logo, the user is taken to the supplier's webshop and is also logged in immediately. The user can then select the desired articles in the usual way and fill the shopping cart.

When the order request is complete, the user is sent back to the Spend Cloud, whereby the shopping cart of the webshop is copied by the Spend Cloud. The order request is not yet placed as an order with the supplier at that time, but will first go through the approval process within the Spend Cloud. The order is automatically offered to the supplier with an order number for processing only after the approval process is completed.



The process looks like this:



How do I implement an OCI connection?

The Spend Cloud uses a connection based on the standard OCI from SAP, a commonly used standard for exchanging catalogs and systems through the "punch-out" technique. Data can be exchanged through this punch-out. For more information about this technique and the realization of connection possibilities, you can consult the [SAP manual](#).

In short, the configuration of an OCI connection requires you to:

1. Make sure the webshop is accessible with a dynamic URL.
2. Redirect your webshop to the Spend Cloud.
3. Ensure shopping cart data and the webshop are compatible.
4. Make sure the webshop can receive orders from the Spend Cloud.
5. Provide Visma | ProActive with the information needed to complete the configuration.

When this project has priority and you have the technical skills, it will usually take about a day. You will find more details below.

Login automatically into the webshop

To access a webshop from the Spend Cloud, it must be possible to access it via a URL, in which the login details of the webshop can be entered dynamically. The Spend Cloud can then provide correct data to the webshop, so the user is automatically logged in.

An example of a URL from supplier Quick Office Supplies:
[https://b2b.quickofficesupplies.nl/invoke/EINL/ocilogin?
USERNAME=\\$username&PASSWORD=\\$password&OCI_VERSION
=3.0&HOOK_URL=\\$hook_url](https://b2b.quickofficesupplies.nl/invoke/EINL/ocilogin?USERNAME=$username&PASSWORD=$password&OCI_VERSION=3.0&HOOK_URL=$hook_url)

Note! **\$username** and **\$password** are variables. These are credentials for logging into the webshop and should be provided by the supplier (Quick Office Supplies). When the username or the password of the user contains special characters, the Spend Cloud will encode them in the URL link. For example the username `purchaser@proactive.nl` will return in the URL as `purchaser%40proactive.nl`

So the username has to be decoded at the supplier's side and we advise not to use special characters.

`$hook_url` is also a variable, but this variable is supplied by the Spend Cloud Procurement module and will need to be used as described in the following paragraph.

Redirect to the Spend Cloud

After the user has compiled the order in the webshop, the articles must be sent to the Spend Cloud Procurement module by a form post. When you automatically log into the webshop, the Spend Cloud indicates to which address the shopping cart should be sent (the hook URL variable). This value needs to be stored by the supplier in the webshop so the contents of the shopping cart can be sent back to the Spend Cloud Procurement module.

In the example of the URL of Quick Office Supplies, the hook URL is shown as boldly marked below:

[https://b2b.quickofficesupplies.nl/invoke/EINL/ocilogin?
USERNAME=\\$username&PASSWORD=\\$password&OCI_VERSION=
3.0&HOOK_URL=**\\$hook_url**](https://b2b.quickofficesupplies.nl/invoke/EINL/ocilogin?USERNAME=$username&PASSWORD=$password&OCI_VERSION=3.0&HOOK_URL=$hook_url)

An example of a hook URL:

`https://mijn.proactive.nl/customer/oci.php?hash=$hash&supplier_id=$supplier_id&administration_id=$administration_id&delivery_address_id=$delivery_address_id`

Multiple variables need to be used when posting the shopping cart back to the Spend Cloud Procurement module.

- **\$hash** is always provided to check if something that is posted originates from the Spend Cloud Procurement module.
- **\$supplier_id** is always provided by the Spend Cloud Procurement module to know from which supplier the shopping cart is.
- **\$administration_id** is only applicable when a customer has different companies/administrations using different webshop accounts. The Spend Cloud Procurement module is required to know from which company the order originates.
- **\$delivery_address_id** is only applicable when a customer has different locations/addresses using different webshop accounts. The Spend Cloud Procurement module is required to know from which location/address the order originates.

Offer shopping cart content

Upon return, the contents of the shopping cart must be offered to the Spend Cloud. Data like the article number, prices, number of items, etc. must be offered using the OCI standard (via a form that is posted). The Spend Cloud bases the order on this information. You can find an example of what this looks like on the next page.



#2



Line total excl. VAT (€) 14,15 · Line total incl. VAT (€) 15,00

Brand	-	Unit	Doos
Description	Bloemen Ziekte € 15,-	Quantity *	1.00
Item number	-	Organizational unit *	
Amount per unit excl. VAT	14.15		
VAT %	0% BTW		
Amount per unit incl. VAT	15.00		

#1



Line total excl. VAT (€) 14,15 · Line total incl. VAT (€) 15,00

Brand	-	Unit	Doos
Description	Bloemen Jubileum € 15,-	Quantity *	1.00
Item number	-	Organizational unit *	
Amount per unit excl. VAT	14.15		
VAT %	0% BTW		
Amount per unit incl. VAT	15.00		

An elaborate description of all the fields that you can provide is described in detail in the [SAP manual](#) under the heading “4.2 Fields and Field Checks”. Below a table that describes the standard field interpretation of the Spend Cloud Procurement module.

Punch-out field	Spend Cloud field	Required
NEW_ITEM-DESCRIPTION	Article name	Yes
NEW_ITEM-VENDORMAT	Article number	Yes
NEW_ITEM-QUANTITY	Quantity	Yes
NEW_ITEM-UNIT	Unit	Yes
NEW_ITEM-VATPERCENTAGE	VAT	Yes
NEW_ITEM-PRICE	Price excl. VAT	Yes
NEW_ITEM-CURRENCY	Currency	Only if customer needs multi-currency
NEW_ITEM-PRICEUNIT	Price per unit	Only if price calculation is set to price per unit

Other fields can be mapped (such as NEW_ITEM-CUST_FIELD1) on the side of Visma | ProActive. Below you will find a few points of attention, which have to be configured and tested on both sides with the Visma | ProActive contact person:



- The prices are preferably given excluding VAT but there is also a possibility to provide the price including VAT.
- The VAT percentage can also be provided in a different field (than NEW_ITEM-VATPERCENTAGE), for example, NEW_ITEM-CUST_FIELD1 or NEW_ITEM_VAT_PCT. The VAT percentage should always be set as an integer without decimals or with a dot as a decimal separator and without a character for percentage. (Example for '21%', it could be either '21' or '21.00').
- If, in addition to items per unit, weight items are also included in the assortment, the weight must be entered in one of the optional fields (such as NEW_ITEM-CUST_FIELD1).
- Depending on what information is needed to correctly process the order, you can include additional information in the remaining optional fields. These fields will be saved with the order in the Spend Cloud and can be added to the order cXML.
- For accounting purposes, it is advised to provide a category or accounting code in one of the free fields. For example a UNSPC-/NZI-code (this code can be used to translate to a ledger for automated accounting). The fields that can be used are for example NEW_ITEM-SHOP-DISCOUNT and NEW_ITEM-MATGROUP (but could be any other field).
- It is possible to charge the user with delivery costs; these costs have to be included as an extra article in the punch-out. All required fields should be filled for delivery costs as well.

Processing the order

After the order request has gone through the approval process in the Spend Cloud Procurement module, the order request can be entered as an order offered to the supplier. Depending on the possibilities of the supplier the order can be offered as follows:

- The order is offered by e-mail to one fixed e-mail address. The order details are included in a PDF file, a cXML file, or both.
- The order is posted as an XML file directly on the suppliers' server via HTTP(s). Preference is given to the standard cXML, but other standards can be discussed with Visma | ProActive.
- The order is offered as an XML file via (s) FTP.

It is preferred that an order file contains all the information required for the supplier when placing the order. It is not desirable that only an order number is provided where the previously saved order will be processed. This is not desirable because, during the approval process in the Spend Cloud, mutations can take place on the original in the webshop compound order. Numbers can change, but articles can also be removed from the original order in the webshop.

If the processing methods do not apply to the supplier, alternatives may be discussed with us.



Updating the order

Updates of the order are not required for a fully functional OCI connection. It might however provide flexibility to the end-user from the customer and to the supplier. When a user wants to change the quantity of the ordered items or the supplier wants to substitute or deposit items, the order needs to be updated.

User updates the order

The workflow is implemented by a second OCI roundtrip when:

- The same OrderID is stored by the supplier and Spend Cloud.
 - Option 1: External OrderID provided in the first punch-out. The External OrderID value should be placed in an array. Like: EXTERNAL_ORDERNUMBER 0 → 1825TESTEXTORDNR
 - Option 2: Spend Cloud OrderID is stored from the cXML.
- The timestamp reflecting the final change date is provided in the first punch-out (format = yyyy/mm/dd). After this date, the user won't be able to change the order anymore. The timestamp is also placed in an array. Like: MODIFY_DEADLINE 0 → 2022-12-09 12:00:00
- The pending order is stored somewhere, so the user can also edit orders that are not approved yet.
- A second OCI roundtrip; the user is redirected to the shopping cart containing the order, based on the OrderID from the URL. The user can modify and resubmit the pending order.

When an order was already sent to the supplier, the new cXML should replace the previous order (based on OrderID).

Supplier updates the order

This workflow is implemented by using our API V2 when:

1. Authentication is established (docs.proactive.nl).
2. For each update, the PUT endpoint of the API V2 is called.

Required fields for each call (view docs.proactive.nl for reference)

- OrderID should be in the endpoint URL (and response).
 - Option 1: External OrderID provided in the punch-out.
 - Option 2: Spend Cloud OrderID which was stored from the cXML sent by Spend Cloud.
 - Order line IDs (modify order line)
 - Option 1: External OrderID provided in the punch-out.
 - Option 2: Spend Cloud OrderID which was stored from the cXML sent by Spend Cloud.
 - Order line IDs (add new order line)
 - Option 1: External OrderID is provided in the API call.
 - Option 2: Leave empty, Spend Cloud will create a new ID.
 - Order line > VAT Percentage (needed to do calculations)
 - Order line > Unit (needed to do calculations)
4. OrderID is provided in the invoice (same as API call) to match.
 5. The update occurs before the invoice is sent to the customer.

What will orders I receive look like?

Example of an XML file you will receive

Via [this link](#), you will download a zip file that contains two XML files. One with the XML-tags ("standard order + order lines") and the other prefilled ("standard order + orderlines.xml prefilled"), based on a placed order. The tags can be manually configured on the side of Visma | ProActive, so all the necessary information to process the order successfully will be sent to the supplier.

Note: All the information the supplier sends in the punch-out can be returned to the supplier.

Below are two screenshots of the XML files. The first uses XML tags and in the second screenshot, you can see how tags are filled. The first screenshot (with XML tags):

```
cXML payloadID="" timestamp="{%date.currenttime.Y-m-d H:i:s%}" xml:lang="nl">
<Header>
  <From>
    <Credential domain="{%supplier.name%}">
      <Identity>{%tool.username%}</Identity>
    </Credential>
  </From>
  <To>
    <Credential domain="">
      <Identity></Identity>
    </Credential>
  </To>
  <Sender>
    <Credential domain="{%administration.officialname%}">
      <Identity></Identity>
    </Credential>
    <UserAgent/>
  </Sender>
</Header>
<Request deploymentMode="{%client.deploymentmode%}">
  <OrderRequest>
    <OrderRequestHeader orderID="{%order.ordernumber%}" orderDate="{%date.currenttime.Y-m-d%}" type="new">
      <Total>
        <Money currency="EUR">{%order.total_amount.US2%}</Money>
      </Total>
      <ShipTo>
        <Address>
          <Name xml:lang="NL">{%administration.billing_name%}</Name>
          <PostalAddress name="">
            <DeliverTo>{%deliveryemployee.name%}</DeliverTo>
            <DeliverTo></DeliverTo>
            <Street>{%deliveryaddress.address%}</Street>
            <City>{%deliveryaddress.city%}</City>
            <PostalCode>{%deliveryaddress.postal_code%}</PostalCode>
            <Country isoCountryCode="{%deliveryaddress.country_code%}">{%deliveryaddress.country%}</Country>
          </PostalAddress>
        </Address>
      </ShipTo>
      <BillTo>
        <Address>
          <Name xml:lang="NL">{%administration.billing_name%}</Name>
          <PostalAddress name="">
            <DeliverTo>{%deliveryemployee.name%}</DeliverTo>
            <DeliverTo></DeliverTo>
            <Street>{%billingaddress.address%}</Street>
            <City>{%billingaddress.city%}</City>
            <PostalCode>{%billingaddress.postal_code%}</PostalCode>
            <Country isoCountryCode="{%billingaddress.country%}">{%billingaddress.country%}</Country>
          </PostalAddress>
        </Address>
      </BillTo>
      <Contact role="endUser">
        <Name xml:lang="NL">{%deliveryemployee.name%}</Name>
        <Email>{%deliveryemployee.emailaddress%}</Email>
      </Contact>
      <Comments xml:lang="NL">{%order.remark%}</Comments>
      <Extrinsic name="Costcenter"></Extrinsic>
    </OrderRequestHeader>
    <ItemOut lineNumber="{%orderline.linenummer%}" requestedDeliveryDate="{%order.delivery_date.Y-m-d%}" quantity="{%orderline.quantity%}">
      <ItemID>
        <SupplierPartID>{%orderline.articlenumber%}</SupplierPartID>
      </ItemID>
      <ItemDetail>
        <UnitPrice>
          <Money currency="EUR">{%orderline.total_amount.US4%}</Money>
        </UnitPrice>
        <Description xml:lang="nl" domain="">{%orderline.articlename%}</Description>
        <UnitOfMeasure>{%orderline.unit%}</UnitOfMeasure>
      </ItemDetail>
    </ItemOut>
  </OrderRequest>
</cXML>
```

Second screenshot (XML tags filled):

```
<cXML payloadID="" timestamp="2020-01-15 09:39:01" xml:lang="nl">
  <Header>
    <From>
      <Credential domain="Office Depot">
        <Identity>OCI </Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="">
        <Identity/>
      </Credential>
    </To>
    <Sender>
      <Credential domain="The Purchasing Company">
        <Identity/>
      </Credential>
      <UserAgent/>
    </Sender>
  </Header>
  <Request deploymentMode="production">
    <OrderRequest>
      <OrderRequestHeader orderID="I.01.20.0001" orderDate="2020-01-15" type="new">
        <Total>
          <Money currency="EUR">304.92</Money>
        </Total>
        <ShipTo>
          <Address>
            <Name xml:lang="NL">Location B</Name>
            <PostalAddress name="">
              <DeliverTo>Pete the Purchaser</DeliverTo>
              <DeliverTo/>
              <Street>Richard Holkade 10</Street>
              <City>Haarlem</City>
              <PostalCode>2033 PZ</PostalCode>
              <Country isoCountryCode="NL">Nederland</Country>
            </PostalAddress>
          </Address>
        </ShipTo>
        <BillTo>
          <Address>
            <Name xml:lang="NL">The Purchasing Company</Name>
            <PostalAddress name="">
              <DeliverTo>Pete the Purchaser</DeliverTo>
              <DeliverTo/>
              <Street>Richard Holkade 9</Street>
              <City>Haarlem</City>
              <PostalCode>2033 PZ</PostalCode>
              <Country isoCountryCode="Nederland">Nederland</Country>
            </PostalAddress>
          </Address>
        </BillTo>
        <Contact role="endUser">
          <Name xml:lang="NL">Pete the Purchaser</Name>
          <Email>pete@tpcc.com</Email>
        </Contact>
        <Comments xml:lang="NL"/>
        <Extrinsic name="Costcenter"/>
      </OrderRequestHeader>
      <ItemOut lineNumber="1" requestedDeliveryDate="" quantity="1.00">
        <ItemID>
          <SupplierPartID>A277317</SupplierPartID>
        </ItemID>
        <ItemDetail>
          <UnitPrice>
            <Money currency="EUR">26.6200</Money>
          </UnitPrice>
          <Description xml:lang="nl" domain="">Alkalinebatterijen Plus Power AA LR6, Ty</Description>
          <UnitOfMeasure>EA</UnitOfMeasure>
          <Classification/>
        </ItemDetail>
      </ItemOut>
    </OrderRequest>
  </Request>
</cXML>
```

What information does Visma | ProActive need?

We need this information to configure the OCI connection:

- The URL where the webshop can be accessed.
- Login details for a test account.
- Depending on the options for processing orders, the email address to which the order can be sent, or the URL where the order can be posted for a test.
- If the order can be posted, an example of an XML file containing various articles must be included. If weight items can also be ordered, please include these in the example.

Based on this information, we will perform a test in consultation with the supplier, whereby the supplier must monitor whether the order has been received correctly. In particular, if the order is posted, it must be checked whether all fields are filled accurately.

When testing has been successfully concluded, we will configure the following:

- Log in details for the customer.
- The e-mail address or the URL where the order can be offered for actual processing (production).

After the go-live, we will add the supplier to a list of suppliers with which an OCI connection has been set up.

Do you have any questions?

If you have any questions regarding this document, please contact the relevant project consultant. You can receive the contact information through the customer who wishes to use the OCI Webshop to make a connection with the project consultant.



www.proactive.nl
+31 (0)23 54 222 99
info@proactive.nl