



Pflichtenheft

Integration OVK Connect in WebCool

Channel Management

Nerse, Nesim [CBC]

Version 1.0 - 17.10.14

Änderungshistorie

Version	Datum	Bearbeiter	Bemerkungen
1.0	17.10.14	Nesim Nerse	Initiale Erstellung

Inhaltsverzeichnis

Einleitung	4
1. Beschreibung des Ist-Zustandes	5
1.1. WebCOOL Angebots-Auftrags-Prozess	5
1.2. Systemübersicht Connect	7
1.3. Connect Angebots-Auftrags-Prozess	8
1.4. WebServices Connect	17
1.4.1. Allgemeine Connect Web-Services	17
1.4.1.1. <i>getDocumentSchemes</i>	17
1.4.1.2. <i>getOVKParticipants</i>	18
1.4.1.3. <i>setPlacements</i>	19
1.4.1.4. <i>setPricelists</i>	20
1.4.1.5. <i>setAdvertisements</i>	21
1.4.1.6. <i>sendPlacements</i>	22
1.4.1.7. <i>sendPriceLists</i>	23
1.4.1.8. <i>sendAdvertisements</i>	24
1.4.2. Connect Web-Services für den Angebots-Auftrags-Prozess	25
1.4.2.1. <i>sendBusinessTransaction</i>	25
1.4.2.2. <i>getDocumentsList</i>	25
1.4.2.3. <i>fetchBusinessTransaction</i>	27
1.4.2.4. <i>confirmFetch</i>	28
1.4.2.5. <i>undoFetch</i>	29
1.4.2.6. <i>getTransactionsList</i>	29
1.4.2.7. <i>getTransactionStatus</i>	31
2. Soll-Konzept	33
2.1. Freigabe Stammdaten für Connect in WebCOOL	33
2.2. WebCOOL Angebots-Auftrags-Prozess mit Connect Integration	35

Einleitung

Das Projekt „OVK Connect“ (kurz: Connect) soll die Kommunikation zwischen Agenturen und Vermarktern im Rahmen des Angebots-Auftrags-Prozesses weitestgehend automatisieren. Hierzu wurden im Projekt bereits folgende Punkte umgesetzt und stehen ab sofort den an Connect teilnehmenden Agenturen und Vermarktern zur Verfügung:

- Der durch Connect automatisiert abzulaufende Workflow für den Angebots-Auftrags-Prozess wurde definiert. Die nötigen WebServices stehen zur Verfügung.
- Die zwischen Agentur und Vermarkter auszutauschenden Daten (im wesentlichen Anfragen, Angebote, Aufträge, Auftragsbestätigungen und Vermarkter-Stammdaten wie Inventare, Werbeformen und Preislisten) wurden inhaltlich abgestimmt und festgelegt. Als Datenformat wurde XML vereinbart. Der Inhalt der einzelnen Dateien ist in XSD-Dateien definiert. Die XMLs werden unter anderem innerhalb des Angebot-Auftrags-Prozess über WebServices ausgetauscht.
- Die technischen Komponenten (Connect Kit & Connect Center) von Connect wurden implementiert und stehen zur Verfügung.

Agenturen und Vermarkter, die an Connect teilnehmen wollen, müssen an Ihrem Agentur- bzw. Vermarktungs-System Erweiterungen und Anpassungen vornehmen, um den Connect Angebots-Auftrags-Prozess in Ihr System zu integrieren. Dabei werden die XML-Daten via Connect bereitgestellt, und die gewünschten Dokumente können von den Agentur- & Vermarkter-Systemen aufgelistet und heruntergeladen werden. Connect kann nicht bestimmen, wer wann welches Dokument herunterlädt. Das muss durch das verarbeitende System (Agentur- bzw. Vermarkter-System) geregelt werden, inklusive einer Zuordnung zum jeweilig verantwortlichen Sales Mitarbeiter.

Das vorliegende Pflichtenheft dient dazu, den Ist-Zustand des WebCOOL¹ Angebots-Auftrags-Prozesses kompakt darzustellen und die nötigen Erweiterungen und Anpassungen in WebCOOL zu beschreiben, die nötig sind um WebCOOL an Connect anzuschließen.

Da nicht alle Vermarkter und Agenturen an Connect teilnehmen werden, gilt es den Angebots-Auftrags-Prozess ohne Connect weiterhin über WebCOOL zu gewährleisten.

Das vorliegende Pflichtenheft kann auch als Referenz-Dokument zur Anbindung von Connect an Vermarkter-Systeme herangezogen werden am Beispiel von WebCOOL. Dementsprechend ist in diesem Dokument die Aufführung von Connect auf die WebServices eingeschränkt, die aus Sicht der Integration von Connect in ein Vermarkter-System gebraucht werden.

¹ WebCOOL ist das Vermarktungs-System von IP Deutschland

1. Beschreibung des Ist-Zustandes

Nachfolgend wird der derzeitige Angebots-Auftrags-Prozess in WebCOOL beschrieben, welcher zwischen dem Vermarkter IP Deutschland und den anfragenden Werbetreibenden/Agenturen stattfindet. Während die meisten internen Prozess-Schritte beim Vermarkter IP Deutschland innerhalb von WebCOOL ablaufen, findet die Kommunikation mit der Agentur über Email bzw. Telefon statt. Ziel ist es die Kommunikation mit Agenturen, welche an Connect teilnehmen, in den WebCOOL-Prozess zu integrieren und somit die Kommunikation über Telefon und Email abzuschaffen.

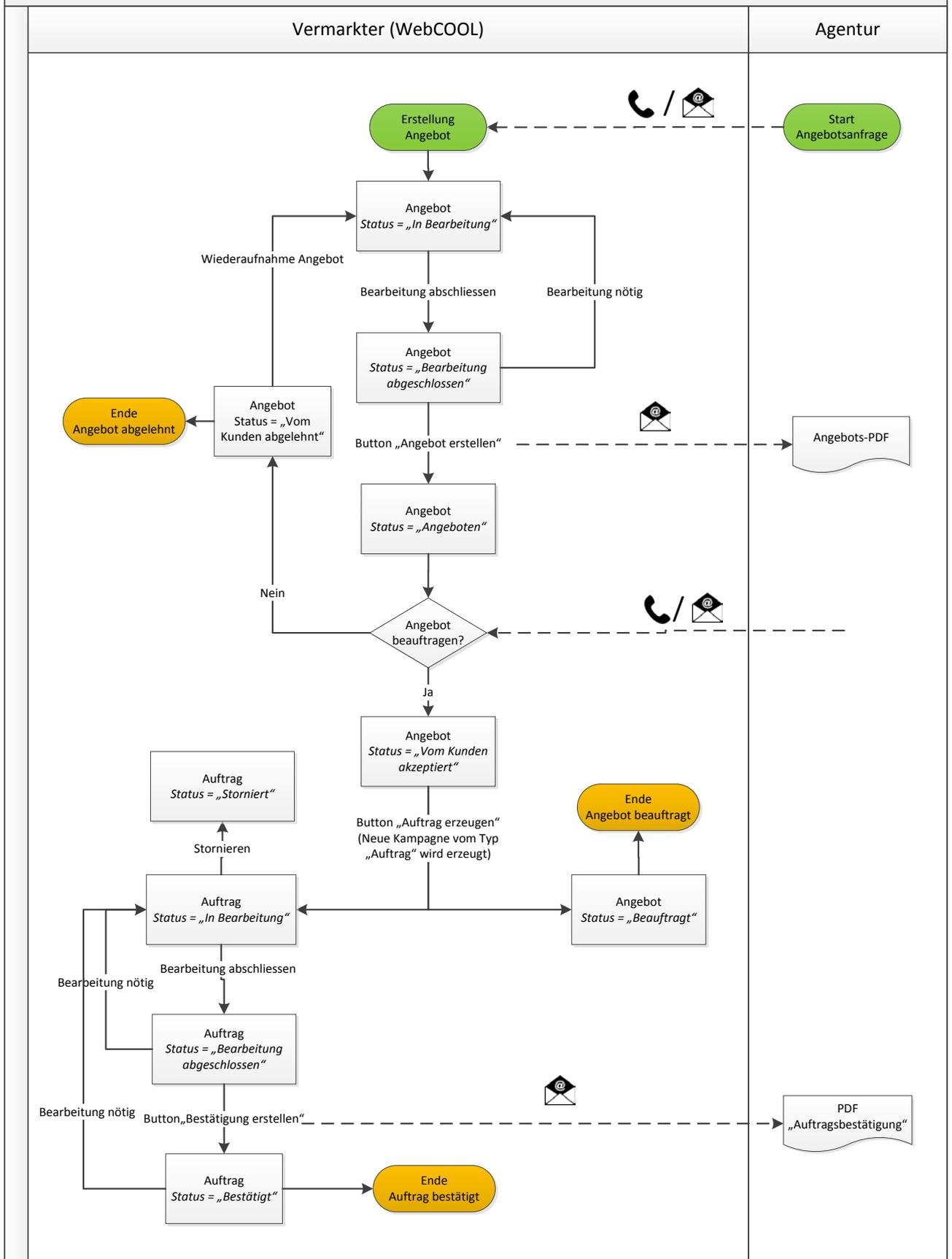
1.1. WebCOOL Angebots-Auftrags-Prozess

Folgendes Workflow-Diagramm veranschaulicht den derzeitigen Angebots-Auftrags-Prozess in WebCOOL. Ziel ist es diesen Prozess an den Connect Angebots-Auftrags-Prozess anzubinden.

Erläuterung: Bei den im Workflow dargestellten Rechtecken handelt es sich um Kampagnen innerhalb von WebCOOL. Beispiel: Eine Kampagne vom Typ „Angebot“ im Status „Bearbeitung abgeschlossen“



Prozessdefinition WebCOOL Angebotsprozess mit AB

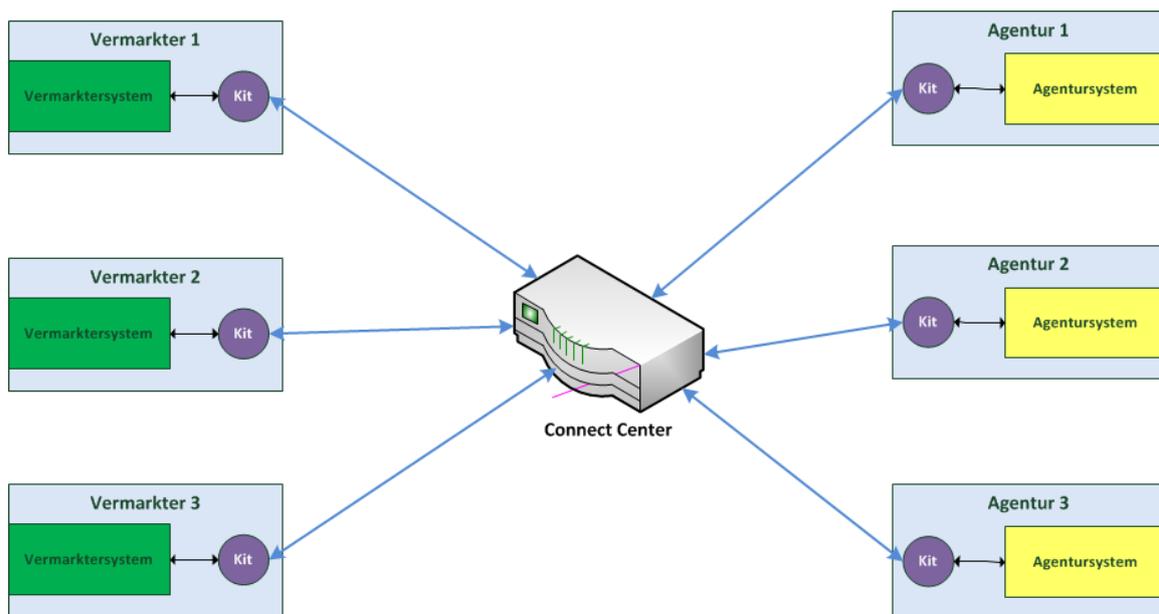


1.2. Systemübersicht Connect

Nachfolgend werden die relevanten Komponenten von Connect vorgestellt, die für die vorgesehene Integration des Connect Angebots-Auftrags-Prozesses in den WebCOOL-Prozess beteiligt sein werden und unbedingt von Nöten sind.

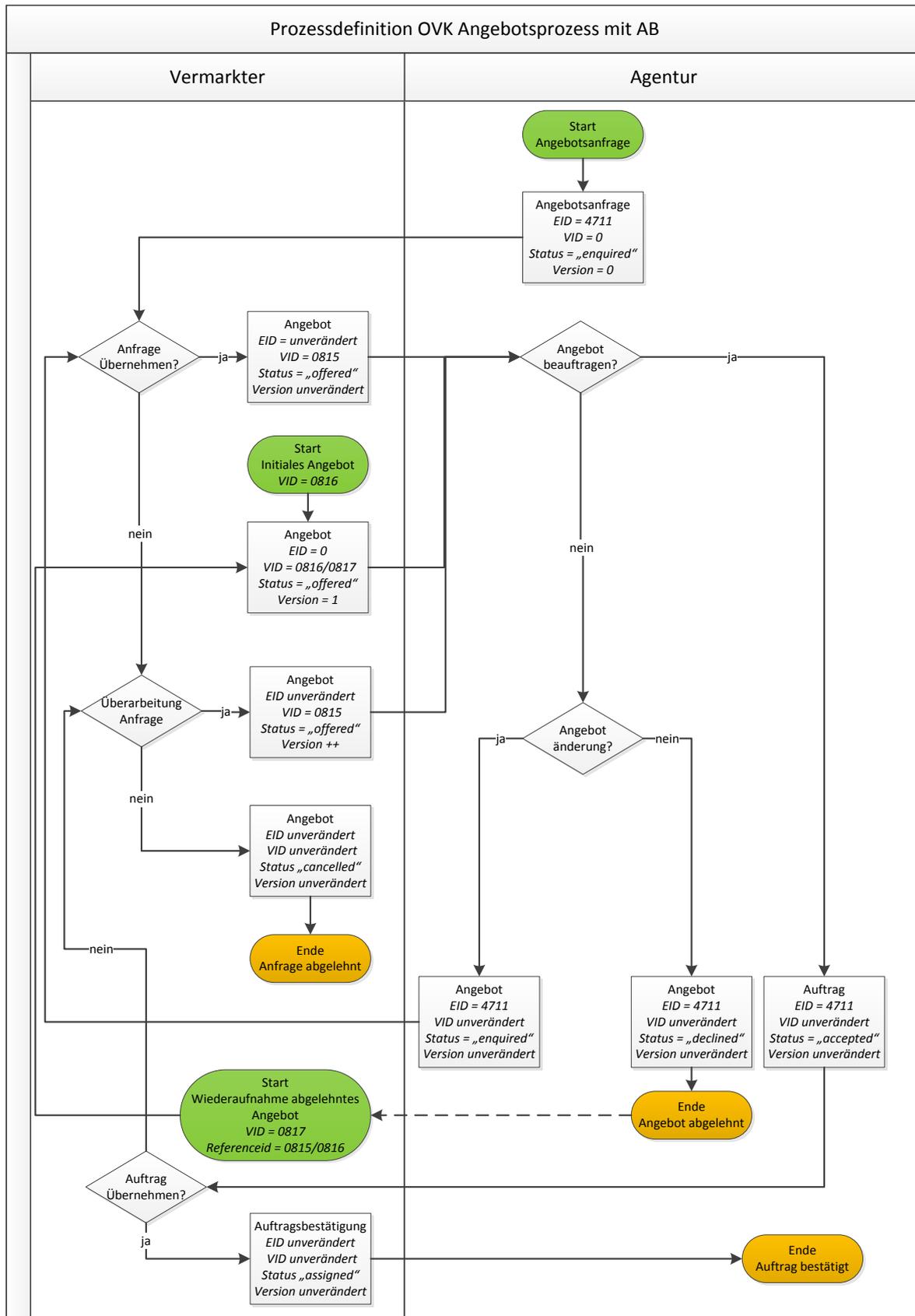
Das zentrale Herzstück ist das Connect Center, ein Webservice, der die Kommunikation zwischen den Teilnehmern abwickelt. Hier werden die kommunizierten Dokumente solange verschlüsselt vorgehalten, bis sie durch den Empfänger abgeholt werden (Pull Verfahren). Dieser Webservice wird durch eine lokale Applikation angesteuert, dem Connect Kit (kurz: Kit). Das Kit wird bei jedem teilnehmenden Akteur (Vermarkter und Agentur) installiert und steuert den genannten Webservice an. Die gesamte Kommunikation des Vermarkter- & Agentur-Systems mit Connect erfolgt also über das Kit.

Wenn beispielsweise das Vermarkter-System ein Angebot an eine Agentur schicken will, muss es das Angebots-XML über den Aufruf eines WebServices, welches von Kit bereitgestellt wird, an die Agentur schicken. Üblicherweise wird das Kit auf einem dedizierten Rechner im Netzwerk des Vermarkters / der Agentur installiert. Folglich würde der Webservice <http://127.0.0.1:8888/api/sendBusinessTransaction> aufgerufen werden, falls der Kit Rechner die IP 127.0.0.1 besitzt.



1.3. Connect Angebots-Auftrags-Prozess

Folgendes Workflow-Diagramm veranschaulicht den Angebots-Auftrags-Prozess über OVK Connect.



Erläuterung: Bei den im Workflow dargestellten Rechtecken handelt es sich um die Bewegungs-Daten, welche durch eine OVK Business-Transaction-XML ausgetauscht werden (definiert durch die XSD-Datei `businesstransaction.xsd`). Eine Business-Transaction-XML kann verschiedene Ausprägungen haben, abhängig davon wie das Type-Element gesetzt ist: Anfrage, Angebot, Auftrag oder Auftragsbestätigung. Beispiel: Eine Business-Transaction vom Typ „Angebot“ im Status „offered“.

Angebot EID = 0 VID = 0816/0817 Status = „offered“ Version = 1
--

Nachfolgend sollen einige zentrale Elemente der Business-Transaction-XML vorgestellt werden:

VID und EID

Einige Elemente der `businesstransaction.xsd` können eine VID und eine EID haben. Die VID ist die ID des jeweiligen Elementes beim Vermarkter und die EID ist die ID bei der Agentur. Die Vermarkter haben die alleinige Hoheit über die Nennung der VID, umgekehrt gilt das gleiche für die Agenturen bei der Nennung der EID. So kann in den verarbeitenden Systemen eine eindeutige, gegenseitige Zuordnung stattfinden. Dieses Konstrukt wurde bewusst gewählt um eine zentrale ID Vergabe zu vermeiden.

Angebotsnummer <offerid>

Die Angebotsnummer kann genau in diesen beiden Ausprägungen, also eine Angebotsnummer des Vermarkters (VID) und der Agentur (EID) identifiziert werden.

Versionsnummer <version>

Die Versionsnummer wird ausschließlich durch den Vermarkter hochgezählt und bietet eine Option für den Angebotsprozess, ein neues korrigiertes Angebot ohne Änderung der Angebotsnummer zu versenden.

Referenz ID <referenceid>

Die Referenz ID dient dazu, einen Bezug zu einem früheren Angebotsprozess herzustellen. So kommt die Referenz ID zum Beispiel zum Einsatz, wenn ein von der Agentur abgelehntes Angebot durch den Vermarkter wiederaufgenommen wird.

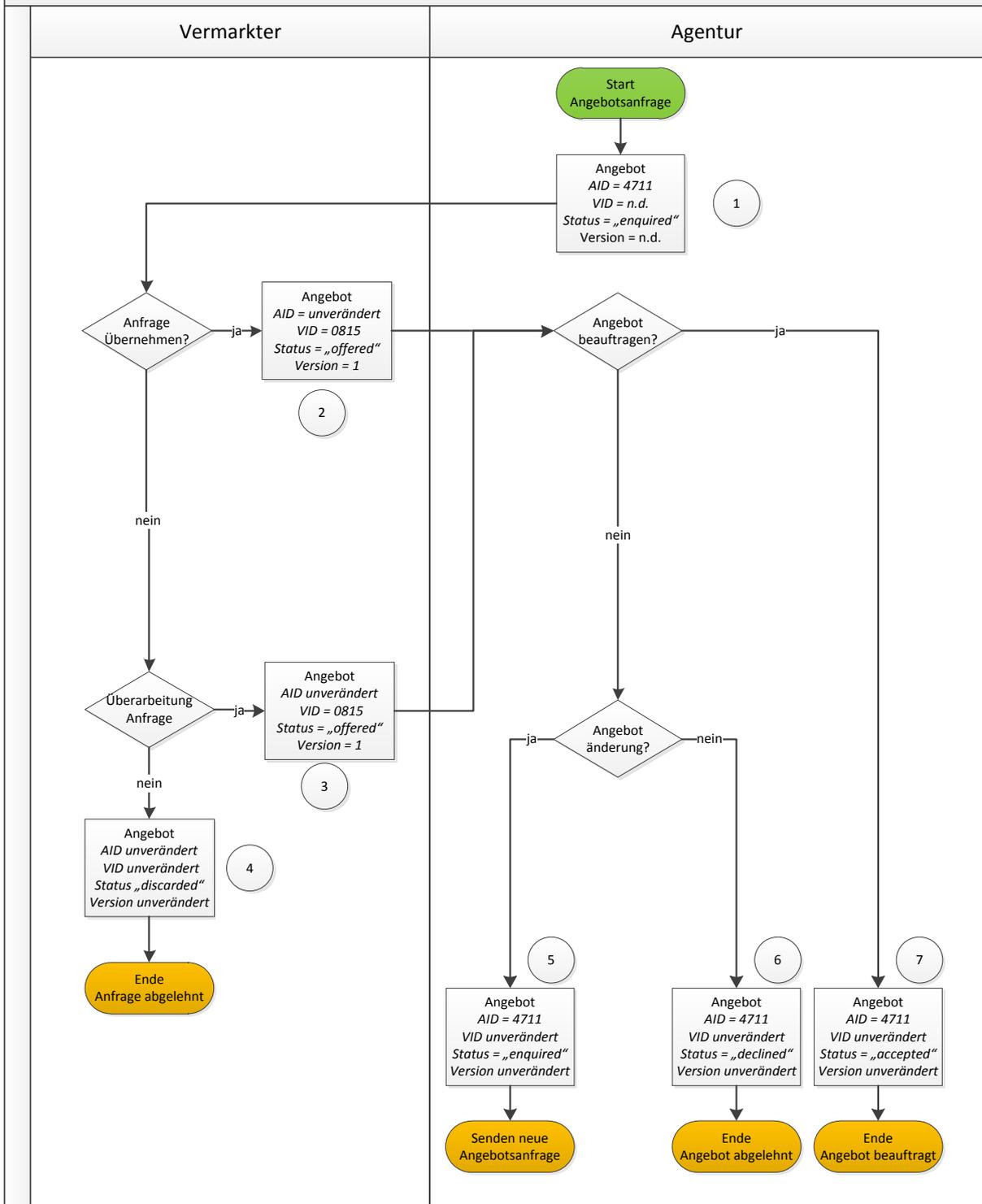
Status <status>

Der Status dokumentiert, in welcher Phase des Angebotsprozesses das generierte Dokument erstellt wurde. Die Phasen sind wie folgt definiert:

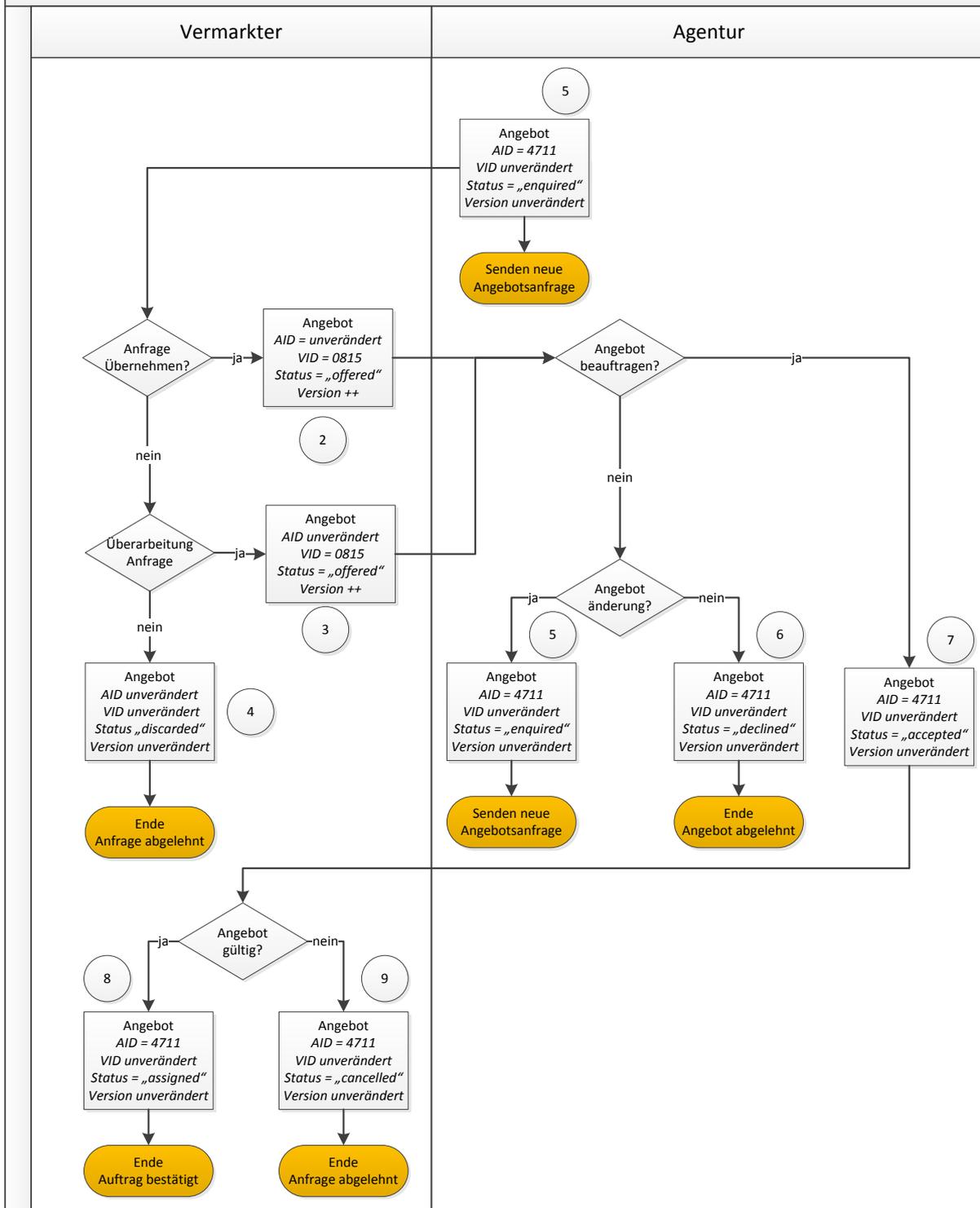
- enquired: Von der Agentur an den Vermarkter angefragt
- offered: Angebot des Vermarkters an die Agentur
- accepted: Agentur akzeptiert das Angebot des Vermarkters (Auftrag)
- declined: Agentur lehnt das Angebot des Vermarkters ab
- assigned: Auftragsbestätigung des Vermarkters
- cancelled: Vermarkter verwirft das Angebot

Nachfolgend wird der Connect Angebots-Auftrags-Workflow beispielhaft durchgegangen.

Prozessdefinition Angebotsprozess



Prozessdefinition Angebotszyklen und Beauftragung



Angebotsanfrage

1

Ebene businesstransaction

Seitens der Agentur besteht die Möglichkeit einer konkreten Angebotsanfrage, welche diese im Namen eines Werbetreibenden anfragt. Die nachfolgende Beschreibung geht auf die Elemente und Attribute ein, die im Rahmen einer Angebotsanfrage eine weitestgehend optimale Integration in das Vermarktungssystem ermöglichen.

Die Agentur gibt über das Attribut <businesstransaction ... EID = „12341“> einen internen Schlüssel mit, der bei der nachfolgenden Kommunikation zwischen Vermarkter und Agentur verwendet werden muss (Klammerkriterium für alle Angebotsversionen). Die VID ist bei einer ersten Anfrage noch nicht bekannt, da diese durch den Vermarkter vorgegeben wird.

```
<businesstransaction VID="32261" EID="234959" ovkparticipantsenderid="28"
ovkparticipantreceiverid="279842">
```

Neben der Referenzierung von VID und EID werden die ovkparticipantids für die sendende und empfangende Seite mitgegeben.

Das Element type wird mit dem Wert „offer“ belegt. - Eine Angebotsanfrage verfügt weder über eine <offer_id> noch eine <version>, da beide durch den Vermarkter vergeben werden. Vielmehr wird die Anfrage mit der Status-Belegung

```
<status>enquired</status>
```

übermittelt. Auch der bei der Agentur verwendete, externe Kampagnenname kann als lesbare Referenz an das Vermarktungssystem übermittelt werden:

```
<externalcampaign>Frühjahr 14</externalcampaign>
```

Es folgt eine Belegung der Participants, also involvierten Teilnehmern, die mit diesem Angebot zu tun haben, was mindestens der werbetreibende Kunde (customer) sein muss. Zur Identifikation von Kunden-Agentur-Vereinbarungen auf Vermarkterseite sollte aber auch die Agentur (agency) mitgeführt werden. Optional können auch Rechnungsempfänger und Ansprechpartner enthalten sein:

```
<participants>
  <customer EID="234707" nielsenID="2374823">
    <name>Testkunde</name>
  </customer>

  <agency ovkparticipantid="28">
    <name>Testagentur</name>
  </agency>

  <company EID="8237" nielsenID="2346872">
    <name>Testagentur Holding</name>
    <type>invoicerecipient</type>
    <address>
      <streetpostbox>Krefelder Straße</streetpostbox>
      <houenumber>67</houenumber>
      <zipcode>40211</zipcode>
      <city>Düsseldorf</city>
      <countrycode>Germany</countrycode>
    </address>
    <contactperson EID="8276">
      <name>Kreditorenbuchhaltung</name>
      <telephone>+49-221-2347-200</telephone>
      <fax>+49-221-2347-299</fax>
    </contactperson>
  </company>
</participants>
```

Wichtig ist auch an dieser Stelle die korrekte Vorbelegung der technischen Schlüssel, um eine automatisierte Übernahme in das Vermarktersystem zu gewährleisten.

Entscheidendes Identifikationsmerkmal bei einem werbetreibenden Kunden ist die NielsenID, da die Stammdaten von OVK Connect keine Participant Informationen beinhalten. Darüber hinaus kann aber auch der technische Schlüssel im Agentursystem in Form der EID mitgegeben werden. Dieser kann nach der ersten Übermittlung dann auch im Vermarktersystem mitgeführt werden, um zukünftige Anfragen auch unabhängig von der NielsenID zuordnen zu können.

Die Agentur kann ausschließlich über die Connect ID (OVKParticipantID) identifiziert werden.

Ebene position

Die Struktur <position> bildet eine Angebots-/Auftrags-)position ab. Über das Attribut @priceltemId wird der entsprechende Preiseintrag der Preisliste (aus den Stammdaten) referenziert. Die @priceltemId referenziert dabei die Kombination aus Inventar, Werbeform und Preis.

Die Agentur kann die Angebotspositionsnummer <posNo> bei der ersten Anfrage vergeben (pro Angebot mit 1 beginnend und danach um 1 inkrementierend). Es ist jedoch zu beachten, dass diese im fortlaufenden Kommunikationsprozess keine eindeutige Identifikation einer Angebotsposition bietet, da beispielsweise Angebotspositionen gelöscht oder zusätzliche hinzugefügt werden können. Diese Zuordnung kann im Bedarfsfall über die VID/EID-Kombination der Struktur <position> erfolgen.

Mittels des Elements referenceid kann die Agentur eine Referenz mitgeben, über welche die Position im Rahmen zukünftiger Prozesse (bis zur Fakturierung) referenziert werden kann. Möchten Vermarkter und Agentur später eine Ermittlung von Zählerdifferenzen durchführen, kann dies über das Element <adcountreferenceid> erfolgen. Diese Elemente können aber auch erst später durch die Agentur ergänzt werden.

```
<referenceid>89203-FGX-23478</referenceid>
<adcountreferenceid>afdd_abf89_7gaa_d9080_gd23</adcountreferenceid>
```

Es folgt die Benennung des angeforderten Inventars über die <placement> Struktur:

```
<placement VID="2484" EID="236478">
  <media VID="1006" EID="73248">
    <name>VOX.de</name>
    <description>DSL Vox, Vox, ptilerecall.de_vox</description>
  </media>
  <name>VOX Homepage</name>
  <description>Homepage</description>
</placement>
```

Hier ist zu beachten, dass eine Referenzierung des Inventars und der Medieninformationen per VID erforderlich ist. Diese müssen im Agentursystem aus den bereitgestellten Stammdaten (placement.xml) des jeweiligen Vermarkters vorgehalten werden.

Die Advertisement Struktur referenziert die Werbeform (die in den Stammdaten per advertisement.xml bereitgestellt wurde). Hier ist neben der VID auch die OVKID (sofern es sich um eine standardisierte Werbeform handelt) ein eindeutiges Identifikationsmerkmal.

```
<advertisement VID="72398" EID="23784" OVKID="2342">
  <name>Homepage Event Advanced (Walltangle)</name>
  <advertisementtype>
    <name>In-Page</name>
  </advertisementtype>
</advertisement>
```

Die Elemente <bookingfrom> und <bookingto> definieren den gewünschten Kampagnenzeitraum der Auftragsposition.

```
<bookingfrom>2014-03-01</bookingfrom>
<bookingto>2014-03-31</bookingto>
```

Mittels des Elements reservationto kann die Agentur einen Reservierungszeitraum für eine Position anfragen

```
<reservationto>2014-02-15</reservationto>
```

Die Price Struktur definiert den Bruttopreislisten Preis (<basicpriceB3>) pro gebuchte Einheit im jeweiligen Preismodell (<standardpricemodel>)

```
<price>
  <basicpriceB3>6000.00</basicpriceB3>
  <pricemodel>
    <standardpricemodel>CPD</standardpricemodel>
  </pricemodel>
</price>
```

Aus der Multiplikation von <basicpriceB3> und <bookedamount> ergibt sich der Wert von <grossB3> in der Struktur <pricecalculation>.

```
<bookedamount>1</bookedamount>
<unit>day</unit>
<pricecalculation>
  <grossB3>6000.00</grossB3>
  <changebasicpriceB3>
    <description>Targeting x 1</description>
    <operation>percental</operation>
    <value>10.00</value>
    <basicpriceB2>600.00</basicpriceB2>
  </changebasicpriceB3>
  <changebasicpriceB3>
    <description>Behavioural Targeting</description>
    <operation>percental</operation>
    <value>5.00</value>
    <basicpriceB2>330.00</basicpriceB2>
  </changebasicpriceB3>
  <grossB2>6930.00</grossB2>
  <grossB1>6930.00</grossB1>
  <customernetN1>6930.00</customernetN1>
  <AE>
    <description>AE</description>
    <value>15.00</value>
    <operation>percental</operation>
    <basePrice>6930.00</basePrice>
    <result>1039.50</result>
    <subtotal>5890.50</subtotal>
  </AE>
  <agencynetN2>5890.50</agencynetN2>
</pricecalculation>
```

Mit dem Element <unit> legt die Agentur fest, welche Einheitenart – z.B. AI, Click, Tag – der angefragten oder angebotenen Menge <bookedamount> zugrunde liegt.

Das Element <sumpricecalculation> und die zugrundeliegende Struktur <pricecalculation> wiederum bilden die Kalkulationsschritte vom B3 (Preislisten Brutto) zum N2 (Agentur Netto) ab.

Über die Positionskomentierung besteht pro Position die Möglichkeit, einen (oder mehrere Begleittexte) an den Vermarkter mitzugeben:

```
<positioncomment commentnr="1">
  Bitte noch auf Commitment 2013 anrechnen
</positioncomment>
```

Angebotslegung



Ebene businesstransaction

Der Vermarkter kann (mit oder ohne vorangegangene Angebotsanfrage) der Agentur ein Angebot übermitteln. Dieses unterscheidet sich in der Belegung der Elemente und Attribute an einigen Stellen zu einer Angebotsanfrage.

```
<businesstransaction VID="32261" EID="234959">
```

Die EID wird mitgeführt, wenn dem Angebot eine Angebotsanfrage vorangegangen ist. Die EID bildet dabei das Merkmal für die Agenturseite, um sämtliche Angebotsversionen zu einer Anfrage zu klammern. Das Element <type> wird mit dem Wert „offer“ belegt. Neben der offer_id wird auch das Status-Element entsprechend gesetzt:

```
<status>offered</status>
```

Mit dem <validto> Element teilt der Vermarkter mit, bis zu welchem Zeitpunkt das Angebot seine Gültigkeit besitzt. Ferner wird optional über die <commitment> Struktur auf die entsprechende Kunde-Agentur-Vereinbarung verwiesen.

```
<validto>2014-02-10</validto> <!-- Gültigkeitsdauer des Angebots -->
<commitment VID="1238" version="2013-01-12">
  <commitment>Kunde-Agentur-Vereinbarung</commitment>
  <version>1</version>
</commitment>
```

Sofern es ein Angebot in der Vergangenheit gibt, auf welches sich dieses Angebot bezieht, kann das frühere Angebot über

```
<referenceid>012123-123-12/2123</referenceid>
```

referenziert werden. Mit der Belegung dieses Feld ist aber keine weitere Verarbeitungslogik verbunden(!). Die Belegung der Participants Elemente wird nun durch den Vermarkter mit den dort geführten Teilnehmerdaten gefüllt.

```
<participants>
  <customer VID="48604" EID="21346" nielsenID="2374823">
    <name>Testkunde</name>
    <type>customer</type>
    <address>
      <streetpostbox>Aachener Str</streetpostbox>
      <houenumber>28</houenumber>
      <zipcode>50679</zipcode>
      <city>Köln</city>
      <countrycode>Germany</countrycode>
    </address>
  </customer>
  <agency VID="47505" ovkparticipantid="28">
    <name>OSE</name>
    <type>agency</type>
    <address>
      <streetpostbox>Postfach 23748</streetpostbox>
      <houenumber />
      <zipcode>40210</zipcode>
      <city>Düsseldorf</city>
      <countrycode>Germany</countrycode>
    </address>
    <contactperson VID="24523" EID="2345">
      <name>Erika Mustermann</name>
      <email>erika.mustermann@agentur.de</email>
      <telephone>+49-221-2387-234</telephone>
      <fax>+49-221-2387-934</fax>
    </contactperson>
    <ntns:ovkparticipantid xmlns:ntns="ovkconnect">28</ntns:ovkparticipantid>
  </agency>
  <company VID="47505" EID="8237">
    <name>Testagentur Holding</name>
    <type>invoicerecipient</type>
    <address>
      <streetpostbox>Krefelder Straße</streetpostbox>
      <houenumber>67</houenumber>
      <zipcode>40211</zipcode>
      <city>Düsseldorf</city>
      <countrycode>Germany</countrycode>
    </address>
    <contactperson VID="34123" EID="8276">
      <name>Kreditorenbuchhaltung</name>
      <telephone>+49-221-2347-200</telephone>
      <fax>+49-221-2347-299</fax>
    </contactperson>
</participants>
```

```
</company>  
</participants>
```

Bei der Übermittlung der participant Informationen ergänzt der Vermarkter seine VIDs (mit den in der Angebotsanfrage übermittelten IDs).

Mit dem Element <earlypaymentdiscount> teilt der Vermarkter der Agentur mit, ob und in welchem Umfang Skonto im Falle einer Buchung gewährt wird.

Ebene position

Sofern es für die die Position eine Reservierungsanfrage gibt, kann der Vermarkter dies durch Setzen des reservationflag vorsehen. Dieser Parameter ist jedoch optional und wird möglicherweise nicht durch alle Vermarkter unterstützt. Durch das Element reservationto wird der Zeitpunkt definiert, bis zu dem die Reservierung vorgenommen wurde.

```
<resevationflag>Y</resevationflag>  
<reservationto>2014-02-10</reservationto>
```

Alle darüber hinaus verwendeten Elemente entsprechen denen der Angebotsanfrage.

Die Frequency Capping Struktur referenziert das Element <fc>, welches per defines.xsd bereitgestellt wird.

Die Targeting Struktur referenziert das Element <targeting>, welches per defines.xsd bzw. targetings.xsd bereitgestellt wird.

Zur Kennzeichnung von Angebotspositionen als rabatt- und skontobildend, kann der Vermarkter die Elemente <discountrelevance> sowie <earlypaymentdiscountrelevance> nutzen.

Mittels den Elementen <guaranteedamount>, <expectedamount> sowie <availableamount> kann der Vermarkter im Falle einer CPM Buchung, sofern möglich und gewünscht, der Agentur mitteilen, welche Menge garantiert ausgeliefert, welche Menge erwartet wird und ob bzw. welche Menge auch als verfügbar zugesichert werden kann.

1.4. Webservice Connect

Die Interaktion mit Connect geschieht über Web-Services. Diese Web-Services werden auf dem [Connect Center](#) bereitgestellt und über den [Connect Kit](#) angesteuert. Nachfolgend werden die für die Integration von Connect in WebCOOL benötigten Web-Services vorgestellt. Diese Web-Services dienen dazu den wesentlichen Connect Angebots-Auftrags-Prozess über Connect im Vermarktungssystem WebCool umzusetzen.

Connect liefert grundsätzlich XML zurück. Der Aufbau ist in den einzelnen API-Aufrufen erläutert. Darüber hinaus existieren folgende allgemeinen XML-Antworten.

ConnectResponse

```
<ConnectResponse status="0" transactionID="2">
  <InfoText>received</InfoText>
</ConnectResponse>
```

Zugehöriges XSD-Schema: [response.xsd](#).

Attribut "status"	0 = OK, 1 = ERROR
Element "InfoText"	Optionaler Infotext für z.B. Erläuterungen bei Fehlerrückgaben
Attribut "transactionID"	In einigen Kommunikationen wird eine TransactionsID erzeugt, welche über dieses Attribut ausgelesen werden kann.

Die Auswertung der Fehlerursache kann über den **HTTP Status Code** ermittelt werden. Die möglichen Fehlercodes sind in den jeweiligen Aufrufen dokumentiert.

Zunächst werden im Kapitel 1.4.1 die allgemeinen Connect Web-Services vorgestellt. Diese dienen dazu die Connect Teilnehmer und XSD-Dateien zu ermitteln. Außerdem werden über die allgemeinen Web-Services die Stammdaten mit den Connect Teilnehmern ausgetauscht.

Im Kapitel 1.4.2 werden dann die Web-Services für den Angebots-Auftrags-Prozess über Connect vorgestellt.

1.4.1. Allgemeine Connect Web-Services

1.4.1.1. *getDocumentSchemes*

Connect liefert grundsätzlich XML zurück. Eine Liste der Strukturen dieser XML-Dateien in unterschiedlichen Versionen ist zu finden unter <https://connectcenterqs01.agof.de/getDocumentSchemes/>

```
<Schemes>
  <BusinessTransaction>
    <Version number="1.4.1" path="https://connectcenterqs01.agof.de/xsd/businesstransacti
on/v1.4.1/BusinessTransaction.xsd" releaseDate="2014-07-04"/>
  </BusinessTransaction>
  <Placements>
    <Version number="1.4.1" path="https://connectcenterqs01.agof.de/xsd/placements/v1.4.1
/Placements.xsd" releaseDate="2014-07-04"/>
  </Placements>
  <Pricelist>
    <Version number="1.4.1" path="https://connectcenterqs01.agof.de/xsd/pricelist/v1.4.1/
Pricelist.xsd" releaseDate="2014-07-04"/>
  </Pricelist>
```

```

<Advertisement>
  <Version number="1.4.1" path="https://connectcenterqs01.agof.de/xsd/advertisement/v1.4.1/Advertisement.xsd" releaseDate="2014-07-04"/>
</Advertisement>
<Defines>
  <Version number="1.4.1" path="https://connectcenterqs01.agof.de/xsd/defines/v1.4.1/Defines.xsd" releaseDate="2014-07-04"/>
</Defines>
</Schemes>

```

Das zugehörige XSD-Schema: [schemelist.xsd](#).

Darüber hinaus existieren folgende allgemeinen XML-Antworten.

ConnectResponse

```

<ConnectResponse status="0" transactionID="2">
  <Info Text>received</InfoText>
</ConnectResponse>

```

Zugehöriges XSD-Schema: [response.xsd](#).

Attribut "status"	0 = OK, 1 = ERROR
Element "InfoText"	Optionaler Infotext für z.B. Erläuterungen bei Fehlerrückgaben
Attribut "transactionID"	In einigen Kommunikationen wird eine TransactionsID erzeugt, welche über dieses Attribut ausgelesen werden kann.

1.4.1.2. *getOVKParticipants*

Jedes an Connect teilnehmende Unternehmen eine OVKParticipant-ID. Diese ID ist für die Kommunikation in *connect* notwendig. Sie identifiziert den jeweiligen Teilnehmer. Eine Liste der vorhandenen OVKParticipants kann man über den Befehl *getOVKParticipants* erhalten werden. Die OVKParticipantID ist in der Rückantwort zu finden. Diese ID ist fix und ändert sich für einen Teilnehmer nicht. Da die OVKParticipantIDs bei der Kommunikation von WebCool mit Connect von essentieller Bedeutung sind (bspw. geben sie Connect vor mit welcher Agentur die Angebote ausgetauscht werden sollen), sollen diese durch WebCool in der WebCool-DB gespeichert (indem der Web-Service aufgerufen wird, siehe auch 2.1).

Aufruf <http://127.0.0.1:8888/api/getOVKParticipants/>

Rückgabe XML-Liste der User (encoding: utf-8)

Beispiel

```

<Users created="2014-7-25 22:10:35">
  <User>
    <ovkparticipantid>2_NOOVK</ovkparticipantid>
    <name>Vermarkter 1</name>
    <type>publisher</type>
    <Schemes>
      <Pricelists>
        <Version>1.2</Version>
      </Pricelists>
      <Placements>
        <Version>1.2</Version>
      </Placements>
    </Schemes>
  </User>
</Users>

```

```

<BusinessTransactions>
  <Version>1.2</Version>
</BusinessTransactions>
<Advertisements>
  <Version>1.2</Version>
...

```

[download](#)

Schema

Das zugehörige XSD-Schema: [OVKParticipants.xsd](#).

Erläuterungen:

Elemente von User		
Element	Attribut	Bedeutung
ovkparticipantid		ID des Users in <i>connect</i> . Diese ID entspricht der OVKParticipantsID .
name		Name des Users im connect center.
type		Der Typ des Users. Es gibt die Typen " publisher " (= Vermarkter) und " user " (= Alle anderen User wie z.B. Agenturen)
Schemas		Dokument Schemas, die der User unterstützt.

1.4.1.3. *setPlacements*

Dient zur Ablage des Inventars eines Vermarkters für den allgemeinen Zugriff. Ein evtl. bereits vorhandenes Inventar des Vermarkters wird überschrieben.

Aufruf `http://127.0.0.1:8888/api/setPlacements`

Methoden POST (**multipart/form-data** encoded)

Rückgabe *ConnectResponse* mit InfoText "received" wenn Empfang abgeschlossen werden konnte.

Parameter

file Die Placements-XML Datei (*multipart/form-data* encoded). Die Dateigröße darf 256 Megabyte nicht überschreiten.

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
400	missing params	Parameter file wurde nicht übergeben.
401	...	Bei einem Authentifizierungsproblem sind grundsätzlich die unter Authentifizierung aufgeführten Antworten möglich.

Code	Infotext	Erläuterung
432		Probleme bei der Validierung der Inhalte

Beispieldatei placements.xml (Ausschnitt):

```
<placements xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://connectcenterqs01.agof.de/xsd/placements/v1.4.2/Placements.xsd" version="2014-06-26" ovkparticipantsenderid="1111">
  <placement VID="22914221" validfrom="2014-01-01">
    <media VID="1007">
      <name>IP-Netzwerk</name>
    </media>
    <name>IP Kochen-Kombi 2014</name>
    <description>Kochen-Kombi</description>
    <availableadvertisement>
      <advertisement VID="57683" compoundadvertisement="false">
        <name>Post-Roll 20 Sek.</name>
      </advertisement>
    </availableadvertisement>
  </placement>
  ...
</placements>
```

Rückgabe

```
<ConnectResponse status="0">
  <InfoText>received</InfoText>
</ConnectResponse>
```

1.4.1.4. setPricelists

Dient zur Ablage der Preislisten eines Vermarkters für den allgemeinen Zugriff. Eine evtl. bereits vorhandene Preisliste des Vermarkters wird überschrieben.

Aufruf <http://127.0.0.1:8888/api/setPricelists>

Methoden POST

Rückgabe *ConnectResponse* mit InfoText "received" wenn Empfang abgeschlossen werden konnte.

Parameter

file *Inhalt der Preislisten-XML Datei (multipart/form-data encoded). Die Dateigröße darf 256 Megabyte nicht überschreiten.*

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
400	missing params	Parameter file wurde nicht übergeben.
401	...	Bei einem Authentifizierungsproblem sind grundsätzlich die unter Authentifizierung aufgeführten Antworten möglich.
432		Probleme bei der Validierung der Inhalte

Beispieldatei pricelist.xml (Ausschnitt):

```
<pricelists xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://connectcenterqs01.agof.de/xsd/pricelist/v1.4.2/Pricelist.xsd" ovkpa
rticipantsenderid="1111" >
  <pricelist issued="2014-06-26" VID="2070" currency="EUR" version="1" validfrom="2013-01-01" validto="201
4-12-31">
    <name>Frauenzimmer Newsletter</name>
    <priceentry VID="53309">
      <placement VID="22914221" validfrom="2014-01-01">
        <media VID="1007">
          <name>IP-Netzwerk</name>
        </media>
        <name>IP Kochen-Kombi 2014</name>
        <description>Kochen-Kombi</description>
      </placement>
      <advertisement VID="57683" compoundadvertisement="false">
    ...
```

Rückgabe

```
<ConnectResponse status="0">
  <InfoText>received</InfoText>
</ConnectResponse>
```

1.4.1.5. setAdvertisements

Dient zur Ablage des Inventars eines Vermarkters für den allgemeinen Zugriff. Ein evtl. bereits vorhandenes Inventar des Vermarkters wird überschrieben.

Aufruf <http://127.0.0.1:8888/api/setAdvertisements>

Methoden POST

Rückgabe *ConnectResponse* mit InfoText "received" wenn Empfang abgeschlossen werden konnte.

Parameter

file Die Advertisements-XML Datei (*multipart/form-data* encoded). Die Dateigröße darf 256 Megabyte nicht überschreiten.

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
400	missing params	Parameter file wurde nicht übergeben.
401	...	Authentifizierungsproblem
432		Probleme bei der Validierung der Inhalte

Beispieldatei advertisements.xml (Ausschnitt):

```
<advertisements xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:noNamespaceSchemaLocation="https://connectcenterqs01.agof.de/xsd/advertisement/v1.4.2/Advertisement.xsd" ovkparticipantsenderid="1111">
  <advertisement VID="57653" compoundadvertisement="false">
    <name>Pre-Roll 20 Sek.</name>
    <advertisementtype VID="1011">
      <name>In-Stream</name>
    </advertisementtype>
    <delivery>
      <destination>banner@ip-deutschland.de</destination>
      <timeframe>5</timeframe>
      <timeframeunit>businessday</timeframeunit>
    </delivery>
  </advertisement>
  ...
```

Rückgabe

```
<ConnectResponse status="0">
  <InfoText>received</InfoText>
</ConnectResponse>
```

1.4.1.6. sendPlacements

Verschickt ein Inventar an eine bestimmte Agentur. Es findet von *connect* aus keinerlei weitere Verarbeitung der verschickten Datei statt. Das *connect* speichert lediglich den Inhalt der Datei. Somit ist es z.B. auch möglich die Datei komprimiert oder verschlüsselt zu übergeben.

Aufruf <http://127.0.0.1:8888/api/sendPlacements/>

Methoden **POST** (*multipart/form-data* encoded)

Rückgabe *ConnectResponse* mit InfoText und der TransactionsID im Attribut

Parameter

receiver Empfänger, ID der Agentur in *connect* (ID kann über *getOVKParticipants* bestimmt werden)

file Inhalt der Inventory-XML Datei (*multipart/form-data* encoded). Die Dateigröße darf 256 Megabyte nicht überschreiten.

attr_XXX Eine beliebige Anzahl von Attributen (key/value pairs) um das Inventory mit Metainformationen anzureichern. Diese haben das Prefix *attr_* gefolgt vom Attributname (z.B. *attr_codeinternal*)

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
400	missing params	Parameter file wurde nicht übergeben
409	unknown receiver	Der angegebene Empfänger ist <i>connect</i> nicht bekannt.
401	...	Authentifizierungsproblem

Code	Infotext	Erläuterung
432		Probleme bei der Validierung der Inhalte

Beispiel

```
<ConnectResponse status="0" transactionID="317">
  <InfoText>Inventory for Agentur 1 received</InfoText>
</ConnectResponse>
```

1.4.1.7. *sendPriceLists*

Verschickt eine Preisliste an eine bestimmte Agentur. Es findet von *connect* aus keinerlei weitere Verarbeitung der verschickten Datei statt. Das *Connect* speichert lediglich den Inhalt der Datei. Somit ist es z.B. auch möglich die Datei komprimiert oder verschlüsselt zu übergeben.

Aufruf <http://127.0.0.1:8888/api/sendPriceLists>

Methoden **POST** (*multipart/form-data* encoded)

Rückgabe *ConnectResponse* mit InfoText und der TransactionsID im Attribut

Parameter

receiver	Empfänger, ID der Agentur in <i>connect</i> (ID kann über getOVKParticipants bestimmt werden)
file	Inhalt der Preislisten-XML Datei (<i>multipart/form-data</i> encoded). Die Dateigröße darf 256 Megabyte nicht überschreiten.
attr_XXX	Eine beliebige Anzahl von Attributen (key/value pairs) um die Preisliste mit Metainformationen anzureichern. Diese haben das Prefix <i>attr_</i> gefolgt vom Attributname (z.B. <i>attr_codeinternal</i>)

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
400	missing params	Parameter file wurde nicht übergeben
409	unknown receiver	Der angegebene Empfänger ist <i>connect</i> nicht bekannt.
401	...	Authentifizierungsproblem
432	siehe	Probleme bei der Validierung der Inhalte

Beispiel

```
<ConnectResponse status="0" transactionID="318">
  <InfoText>Pricelists for Agentur 1 received</InfoText>
</ConnectResponse>
```

1.4.1.8. sendAdvertisements

Verschickt die Werbeformen eines Vermarkters an eine bestimmte Agentur. Es findet von *connect* aus keinerlei weitere Verarbeitung der verschickten Datei statt. Das *connect* speichert lediglich den Inhalt der Datei. Somit ist es z.B. auch möglich die Datei komprimiert oder verschlüsselt zu übergeben.

Aufruf <http://127.0.0.1:8888/api/sendAdvertisements>

Methoden **POST** (*multipart/form-data* encoded)

Rückgabe *ConnectResponse* mit InfoText und der TransactionsID im Attribut

Parameter

receiver	Empfänger, ID der Agentur in <i>connect</i> (ID kann über getOVKParticipants bestimmt werden)
file	Inhalt der Werbeformen-XML Datei (<i>multipart/form-data</i> encoded). Die Dateigröße darf 256 Megabyte nicht überschreiten.
attr_XXX	Eine beliebige Anzahl von Attributen (key/value pairs) um die Werbeform mit Metainformationen anzureichern. Diese haben das Prefix <i>attr_</i> gefolgt vom Attributname (z.B. <i>attr_codeinternal</i>)

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
400	missing params	Parameter file wurde nicht übergeben
409	unknown receiver	Der angegebene Empfänger ist <i>connect</i> nicht bekannt.
401	...	Authentifizierungsproblem
432		Probleme bei der Validierung der Inhalte

Beispiel

```
<ConnectResponse status="0" transactionID="319">
  <InfoText>Advertisements for Agentur 1 received</InfoText>
</ConnectResponse>
```

1.4.2. Connect Web-Services für den Angebots-Auftrags-Prozess

1.4.2.1. *sendBusinessTransaction*

Verschickt ein Angebot.

Aufruf <http://127.0.0.1:8888/api/sendBusinessTransaction>

Rückgabe *ConnectResponse* mit InfoText und der TransactionsID im Attribut

Parameter

receiver	Empfänger, ID des Teilnehmers in <i>connect</i> (ID kann über getOVKParticipants bestimmt werden)
file	Inhalt der Angebots-XML Datei (<i>multipart/form-data</i> encoded). Die Dateigröße darf 256 Megabyte nicht überschreiten.
attr_XXX	Eine beliebige Anzahl von Attributen (key/value pairs) um das Angebot mit Metainformationen anzureichern. Diese haben das Prefix <i>attr_</i> gefolgt vom Attributname (z.B. <i>attr_codeinternal</i>)

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
400	missing params	Parameter file wurde nicht übergeben
409	unknown receiver	Der angegebene Empfänger ist <i>connect</i> nicht bekannt.
401		Authentifizierungsproblem
432		Probleme bei der Validierung der Inhalte

Beispiel

```
<ConnectResponse status="0" transactionID="2-479">
  <InfoText>BusinessTransaction for Vermarkter 2 received</InfoText>
</ConnectResponse>
```

1.4.2.2. *getDocumentsList*

Gibt eine Liste der von Vermarktern/Agenturen für den eingeloggten User eingestellten Angebote und anderer Inhalte aus (siehe bspw. [sendBusinessTransaction](#)), deren Abruf noch nicht mit [confirmFetch](#) bestätigt wurde. Kann auf die Angebote eines Vermarkters eingeschränkt werden.

Die Angebote werden dabei nicht abgerufen. Jedes Angebot hat ein **id** Tag, das es eindeutig identifiziert. Diese **id** muss beim eigentlichen Abruf (siehe [fetchBusinessTransaction](#)) genutzt werden.

Aufruf <http://127.0.0.1:8888/api/getDocumentsList>

Rückgabe XML-Liste der Angebote (encoding: utf-8)

Parameter	
type	Typ der Angebote, " businesstransaction " für Angebote (siehe sendBusinessTransaction), " placements " für Inventare (siehe sendPlacements), " pricelist " für Preislisten (siehe sendPriceLists) oder " advertisement " für Werbeformen (siehe sendAdvertisements). Parameter ist optional. Keine Angabe gibt alle Typen zurück.
sender	ID des Vermarkters, der das Angebot (bzw. die Angebote) eingestellt hat (ID kann über getOVKParticipants bestimmt werden)
lastChanged	Angabe eines Datumsbereiches (von-bis im Format " <i>JJJJ-MM-DD, JJJJ-MM-DD</i> "). Das erste Datum definiert den Beginn des Bereichs, das Zweite (optionale) das Ende. Der Parameter ist optional. Beispiele: " <i>2014-01-01</i> ", " <i>2014-01-01, 2014-02-01</i> ". Neben dem axakten Datumsformat ist auch " TODAY " als Datumsangabe möglich. Bei diesem Format können beim Bereichsende zur Zahlen angegeben werden. Beispiele: " <i>TODAY</i> " => alles von heute, " <i>TODAY,-1</i> " => alles seid gestern, " <i>TODAY,-2</i> " => alles seid vorgestern

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
401		Authentifizierungsproblem

Beispiel

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Documents>
  <Document id="289" date="2013-08-09 11:12:24.0" >
    <Sender id="2" name="Vermarkter 1" />
    <Attributes>
      <Attribute key="pw" value="test" />
      <Attribute key="encrypted" value="1" />
      <Attribute key="type" value="offer" />
      <Attribute key="waitForConfirm" value="0" />
    </Attributes>
  </Document>
  <Document id="290" date="2014-02-22 14:29:24.0" >
    <Sender id="3" name="Agentur 1" />
    <Attributes>
      <Attribute key="type" value="offer" />
      <Attribute key="waitForConfirm" value="0" />
    </Attributes>
  </Document>
  <Document id="291" date="2014-04-17 15:27:22.0">
    <Sender id="3" name="Agentur 1" />
    <Attributes>
  ...
```

Zugehöriges XSD-Schema: [documentsList.xsd](#).

Elemente von "Document"

Element	Attribut	Bedeutung
	id	Die TransactionsID des Dokumentes. Eindeutige ID im <i>connect</i> System.
	date	Zeitpunkt zu dem das Dokument eingestellt wurde im YYYY-MM-DD HH:MM:SS Format
Sender		Der Sender des Dokumentes
Sender	id	ID des Users in <i>connect</i> . Diese ID entspricht der OVKParticipantsID .
Sender	name	Name des <i>connect</i> users
Attributes		Sammlung von META-Informationen zum Dokument. Hier stehen technisch vom System vergebene Informationen wie z.B. der Typ des Dokumentes als auch vom Sender beim Versand angegebene Attribute.
Attribute	key, value	Ein Attribut ist durch ein Key/Value Paar gekennzeichnet.

1.4.2.3. *fetchBusinessTransaction*

Ruft ein Angebot aus dem connect center ab. Der *Fetch* erfolgt in zwei Stufen. Wenn der Fetch erfolgreich war, muss er im Anschluss mit *confirmFetch* abgeschlossen werden, um den Inhalt im connect center dauerhaft zu entfernen. Wenn ein Fetch nicht funktioniert hat, kann man nach Aufruf von *undoFetch* den Inhalt erneut abrufen.

Aufruf <http://127.0.0.1:8888/api/fetchBusinessTransaction>

Rückgabe Eingestelltes Angebot

Parameter

id	Angebots-ID (id aus <i>getDocumentsList</i>)
-----------	---

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
400	missing params	Angebots-ID wurde nicht übergeben.
404	not found	Das Angebot unter der ID konnte nicht gefunden werden.
409	already fetched	Das Angebot wurde bereits abgerufen, ggf. <i>undoFetch</i> nutzen.
401		Authentifizierungsproblem
432		Probleme bei der Validierung der Inhalte

Beispiel

```
<businesstransaction xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance xsi:noNamespaceSchemaLocation="https://connectcenterqs01.agof.de/xsd/businesstransaction/v1.4.2/BusinessTransaction.xsd" VID="23103278" version="2014-06-26" ovkparticipantsenderid="1111" ovkparticipantreceiverid="1234">
  <type>offer</type>
  <offerid>42239</offerid>
  <version>1</version>
  <status>offered</status>
  <validto>2014-11-23</validto>
  <campaign>Finanzplanung privat Test </campaign>
  <participants>
    <advertiser VID="71190">
      <name>Beispiel Giroverband e.V.</name>
      <type>advertiser</type>
      <address>
        ...
      </address>
    </advertiser>
  </participants>
</businesstransaction>
```

1.4.2.4. confirmFetch

Wenn Angebote aus dem connect center abgerufen werden (siehe [fetchBusinessTransaction](#)) dann werden diese nicht gelöscht. Um diese Angebote zu löschen muss man die Abholung der Angebote mit den Aufruf dieser Funktion bestätigen. Erst dann wird das abgerufene Angebot gelöscht.

Aufruf <http://127.0.0.1:8888/api/confirmFetch>

Rückgabe *ConnectResponse* mit InfoText "fetch confirmed" oder "nothing to confirm"

Parameter

id	id aus getDocumentsList
----	---

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
400	missing params	Parameter id wurde nicht übergeben.
401	...	Authentifizierungsproblem

Beispiel

```
<ConnectResponse status="0" transactionID="316">
  <InfoText>fetch confirmed</InfoText>
</ConnectResponse>
```

1.4.2.5. *undoFetch*

Wenn beim Abruf von Angeboten durch *fetchBusinessTransaction* aus dem connect center Fehler auftreten, dann kann man mit diesen Befehl die Abrufung des Angebotes rückgängig machen. Erst dann kann man das Angebot erneut abrufen.

Aufruf <http://127.0.0.1:8888/api/undoFetch>

Rückgabe *ConnectResponse* mit InfoText "undo fetch" oder "nothing to undo"

Parameter

id	id aus <i>getDocumentsList</i>
-----------	---------------------------------------

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
400	missing params	Parameter <i>id</i> wurde nicht übergeben.
401	...	Authentifizierungsproblem

Beispiel

```
<ConnectResponse status="0" transactionID="316">
  <InfoText>undo fetch</InfoText>
</ConnectResponse>
```

1.4.2.6. *getTransactionsList*

Gibt eine Liste der für andere eingestellten Dokumente und deren Status aus.

Status	Erläuterung
1	eingestellt => Das Dokument wurde eingestellt, aber noch nicht abgerufen.
2	abgerufen => Das Dokument wurde abgerufen, aber der Abruf wurde noch nicht mit <i>fetch</i> bestätigt.
3	bestätigt => Das Dokument wurde abgerufen und der Empfang per <i>fetch</i> bestätigt.

Aufruf [http:// 127.0.0.1:8888/api/getTransactionsList](http://127.0.0.1:8888/api/getTransactionsList)

Methoden **GET**

Rückgabe XML-Liste der Angebote (encoding: utf-8)

Parameter

states	Einschränkung auf den Status der Dokumente. Mögliche StatusIDs siehe oben. Es können mehrere States angegeben werden (kommasepariert). Parameter ist optional. Beispiel: "1,3"
lastChanged	Angabe eines Datumsbereiches (von-bis im Format "JJJJ-MM-DD, JJJJ-MM-DD"). Das erste Datum definiert den Beginn des Bereichs, das Zweite (optionale) das Ende. Der Parameter ist optional. Beispiele: "2014-01-01", "2014-01-01, 2014-02-01".

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
400	wrong param format	Eines der beiden Datumsangaben im lastChanged Parameter hatte ein ungültiges Format.
401	...	Authentifizierungsproblem

Beispiel

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Documents>
  <Document id="4">
    <Receiver id="3" name="Agentur 1" />
    <State id="3" info="bestätigt" changed="2013-01-21 16:46:10.0" />
    <Attributes>
      <Attribute key="campain" value="WSV2013" />
      <Attribute key="internalid" value="1234" />
      <Attribute key="type" value="useroffer" />
    </Attributes>
  </Document>
  <Document id="63">
    <Receiver id="3" name="Agentur 1" />
    <State id="3" info="bestätigt" changed="2013-03-06 19:26:15.0" />
    <Attributes>
      <Attribute key="type" value="useroffer" />
    </Attributes>
  </Document>
</Documents>
```

Zugehöriges XSD-Schema: [documentinfoList.xsd](#).

Elemente von "Document"

Element	Attribut	Bedeutung
	id	Die TransactionsID des Dokumentes. Eindeutige ID im <i>connect</i> System.
Receiver		Der Empfänger des Dokumentes
Receiver	id	ID des Users in <i>connect</i> . Diese ID entspricht der OVKParticipantsID .
Receiver	name	Name des <i>connect</i> users

Element	Attribut	Bedeutung
State		Der Status des Dokumentes.
State	id	Status ID.
State	name	Name für den Status.
State	changed	Zeitpunkt zu dem der aktuelle Status des Dokumentes sich geändert hat
Attributes		Sammlung von META-Informationen zum Dokument. Hier stehen technisch vom System vergebene Informationen wie z.B. der Typ des Dokumentes als auch vom Sender beim Versand angegebene Attribute.
Attribute	key, value	Ein Attribut ist durch ein Key/Value Paar gekennzeichnet.

1.4.2.7. *getTransactionStatus*

Gibt den Status eines einzelnen Dokumentes zurück. Erläuterung Status siehe *getTransactionsList*.

Aufruf <http://127.0.0.1:8888/api/getTransactionStatus>

Methoden **GET**

Rückgabe XML-Liste der Angebote (encoding: utf-8)

Parameter

id	ID des Dokuments. Die ID entspricht der TransactionID, die beim Einstellen von Dokumenten zurückgegeben wird (siehe z.B. Rückgabe bei sendBusinessTransaction). Die ID wird auch bei getTransactionsList ausgegeben.
-----------	---

Mögliche Fehlerrückgaben

Code	Infotext	Erläuterung
400	missing params id	Der Parameter <i>id</i> wurde nicht übergeben.
400	document not found	Unter der angegebenen ID wurde kein Dokument gefunden.
401	...	Authentifizierungsproblem

Beispiel

```
<Document id="200">
  <Receiver id="3" name="Agentur 1" />
  <State id="3" info="bestätigt" changed="2013-08-09 11:12:24.0" />
  <Attributes>
    <Attribute key="pw" value="test" />
    <Attribute key="encrypted" value="1" />
    <Attribute key="type" value="offer" />
  </Attributes>
</Document>
```

Zugehöriges XSD-Schema: [documentinfo.xsd](#), Erläuterung hierzu siehe: `getTransactionList`.

2. Soll-Konzept

In diesem Kapitel wird auf den Soll-Zustand eingegangen, der durch die Integration von Connect in WebCOOL entstehen soll.

2.1. Freigabe Stammdaten für Connect in WebCOOL

Zunächst sollen die Erweiterung in WebCOOL beschrieben werden, welche eine Voraussetzung für den „Connect Angebots-Auftrags-Prozess über WebCOOL“ darstellen. Hierbei handelt es sich um die allgemeine und Agentur-spezifische Freigabe von Stammdaten in WebCOOL. Nur durch diese Freigabe können die Stammdaten an das Connect Center, und somit an die Connect-Teilnehmer veröffentlicht werden. Anhand dieser veröffentlichten Stammdaten können Agenturen Anfragen an IP Deutschland stellen.

Die Stammdaten (Inventare, Werbeformen, Preislisten) können derzeit bereits über WebCOOL für OVK Connect freigeschaltet werden.

Die OVK Connect Freigabe bewirkt hierbei, dass die Stammdaten an alle OVK Connect Agenturen veröffentlicht werden. Diese Freigabe der Stammdaten muss erweitert werden, sodass es über WebCOOL auch möglich ist, Stammdaten für spezielle Agenturen über OVK Connect freizuschalten.

Die kundenspezifische Freigabe von Stammdaten ist zwingend erforderlich, da über Connect Vermarkter-Stammdaten, neben der Möglichkeit an alle OVK Agenturen veröffentlicht zu werden, auch speziellen Agenturen veröffentlicht werden sollen. Die Liste der OVK Agenturen und Vermarkter kann durch den Aufruf des [WebServices getOVKParticipants](#) ermittelt werden. WebCOOL sollte die Möglichkeit bieten diese Liste zu importieren, die importierten OVK Connect Teilnehmer (Agentur/Vermarkter) zu speichern und dem User eine Zuordnung der Agenturen aus WebCOOL zu den importierten OVK Connect Agenturen anbieten. Der Inhalt dieser Liste sieht wie folgt aus:

```
<Users created="2014-10-13 15:35:0">
  <User>
    <ovkparticipantid>2_NOOVK</ovkparticipantid>
    <name>Vermarkter 1</name>
    <type>publisher</type>

    <Schemes>
      <Pricelists>
        <Version>1.2</Version>
      </Pricelists>
      <BusinessTransactions>
        <Version>1.2</Version>
      </BusinessTransactions>
      <Advertisements>
        <Version>1.2</Version>
      </Advertisements>
      <Placements>
        <Version>1.2</Version>
      </Placements>
    </Schemes>
  </User>
</Users>
```

Hierbei ist zu beachten, dass Vermarkter den <type> „publisher“ haben und Agenturen den <type> „user“ besitzen. In dieser Liste finden sich unter anderem die OVKParticipantID's der Vermarkter und Agenturen, welche für den OVK Angebots-Auftrags-Prozess unbedingt von Nöten sind.

Durch die Zuordnung einer Connect Agentur zu einer WebCOOL-Agentur werden alle entsprechenden Daten aus der OVK Teilnehmer Liste der WebCOOL-Agentur zugeordnet. Hierbei handelt es sich unter anderem um die OVKParticipantID, welche für den Connect Angebots-Auftrags-Prozess unbedingt von Nöten ist. Aber auch bei der Veröffentlichung von Stammdaten an spezielle Agenturen wird diese OVKParticipantID benötigt. Die Veröffentlichung der Stammdaten an spezielle Kunden sollte über einen neuen Button in WebCOOL möglich sein.

Nach Betätigung des Buttons, sollte eine Auswahl an Connect Agenturen angeboten werden, an welche die Stammdaten veröffentlicht werden sollen. Nach Auswahl der Agenturen werden die für die jeweilige Agentur/-en freigeschalteten Stammdaten über den Aufruf der Webservice *sendPlacements*, *sendPricelists* *sendPlacements* mit entsprechender Belegung der Variablen an die einzelnen OVK Connect Agentur/-en übermittelt.

Hinweis: Hierbei müssen u.U. mehrere Webservice-Aufrufe für die einzelnen Stammdaten-Typen (Inventar, Werbeformen, Preislisten) und Agenturen stattfinden. Die Rückmeldungen der Webservice-Aufrufe müssen dem WebCOOL-User in geeigneter Weise angezeigt und in das Protokoll-Modul von WebCOOL aufgenommen werden.

2.2. WebCOOL Angebots-Auftrags-Prozess mit Connect Integration

Folgendes Workflow-Diagramm stellt den WebCOOL Angebots-Auftrags-Prozess mit Connect-Anbindung dar. Die im Workflow dargestellten Daten innerhalb der Bewegungsdaten stellen Connect-Daten (gekennzeichnet durch „(OVK)“) und WebCOOL-Daten (gekennzeichnet durch „(WebCOOL)“) dar.

