Typisches Anbindungszenario am Beispiel einer einfachen Soap-Server/Client-Umgebung

Prämisse

Die Warenwirtschaft ist an einen Webserver angebunden. Auf diesem Webserver befindet sich ein Soapserver, der bestimmte Methoden / Klassen der Warenwirtschaft abstrahiert und einem Webclient in Form einer WSDL zur Verfügung stellt. Angebunden ist der Webserver mindestens mit einer VDSLLeitung mit einem Upstream von 2 Mbit. Der Shopserver selbst ruft die WSDL ab, bildet aufgrund dessen eine (PHP)-Klasse und führt per Remote-Aufruf bestimmte Methoden auf. Die folgenden Punkte enthalten Dummy-Code, der in etwa beschreibt, wie wir die Methoden aufrufen und welchen Response wir erwarten.

Reguläre Methoden

Preisfindung, Lagerbestandsanzeige

Preis- und Lagerabfrage sind im Idealfall gebündelt. Wir übergeben eine Anfrage in Form von:

soapclient.php

```
$menge = array('4711' => '1', '0815' => '10', '12345' => '4','1100101'
=> 8)
$artikelliste = array('4711','0815','12345','1100101');
$resultset =
CLIENT::get_preis($artikelliste,$mengenliste,$kundennummer);
```

Das $resultset ist ein Array, dass wie folgt aufgebaut ist:

results.php

```
$resultset['4711']['NETTO'] = '100';
$resultset['4711']['BRUTTO'] = '119';
$resultset['4711']['AVAIL'] = '10';
$resultset['4711']['ORIGINAL'] = 200;
$resultset['4711']['RABATT'] = '50';
$resultset['0815']['NETTO'] = '10';
$resultset['0815']['BRUTTO'] = '19';
$resultset['0815']['AVAIL'] = '3';
$resultset['0815']['ORIGINAL'] = 10;
$resultset['0815']['RABATT'] = false;
```

Es handelt sich hier IMMER um den bereits fertig kalkulierten BESTPREIS für den Kunden. Minimal benötigen wir den Netto/Bruttopreis, sowie den AVAIL-Wert. Der Primärschlüssel ist die Artikelnummer, damit wir innerhalb unserer Shopfunktionen wieder auf die Artikel zurück mappen können.

Bestellübergabe

Die Bestellung selbst ist i.d.R. nur ein Warenkorb mit Kopfdaten. Wir übergeben hier eine Artikelliste und Adressfelder.

order.php

```
$resultset = CLIENT::DO_ORDER()
$lieferadresse['name'] = Name;
$lieferadresse['firma'] = Firma;
$lieferadresse['strasse'] = strasse;
```

Das gleiche gilt für eine Rechnungsadresse.

Anschließend werden die Positionsdaten mit übertragen, dies ist auch „nur“ eine Liste der Artikelnummern PLUS der zugehörigen Menge. Zuletzt kann noch eine „Fake“-Nummer für Versandkosten übergeben werden, oder aber der Webservice bietet direkt ein Feld, das solch einen Wert abfangen kann. Als Resultset erwartet der Shop lediglich die Belegnummer des Auftragsformulars. Diese wird im Shop der Shop-Bestellung zugemapped. Wenn keine Nummer zurück kommt, können wir die Bestellung durch den Admin per Button noch einmal auslösen lassen.

Belegauskunft

Hier muss erst einmal geklärt werden, welche Belegarten der Kunde überhaupt darstellen möchte. Technisch sollte die Abfrage aber immer relativ eindeutig sein. Zuerst gibt es immer eine Suche.

Suche nach Zeitraum

beleg.php

```
$resultset =
CLIENT::Belegliste('Angebot','1.1.2014','31.3.2014',kundennummer,false)
;
```

bzw. Suche nach eindeutiger Nummer

beleg2.php

```
$resultset =
```

```
CLIENT::Belegliste('Angebot','1.1.2014','31.3.2014',kundennummer,'BELEG
NUMMER');
```

Das Resultset sollte immer eine Liste mit Belegen sein, auf die dieser Suchfilter passt. Im Idealfall besteht das Resultset aus folgendenen Daten: belegnummer, Positionsdaten (Artikelpositionen des Beleges), Belegsumme, Sachbearbeiter. Diese Daten können wir dann z.B. weiter verwenden, wenn wir ein Angebot direkt wieder in die Bestellung übergeben wollen. Für die eigentliche Darstellung eines Belegausdrucks brauche wir nur die eigentliche Belegnummer.

pdfdownload.php

```
$pdf = CLIENT::get_beleg(nummer, kundennummer);
```

Das $pdf ist ein base64-encodiertes PDF, das wir lediglich dekodieren und dann 1:1 an den User ausgeben.

Weitere technische Gegebenheiten

Webservices sprechen untereinander am besten UTF8. Hier gibt es die größte, handelsübliche Menge an Zeichen, die man benötigt. Insbesondere Umlaute sind immer wieder ein Problem, dass man umgehen kann, wenn von Anfang an auf Unicode gesetzt wird.

Non-Live-Daten

Artikelnummern

Die Artikelnummern des Shops müssen mit den Artikelnummern aus der Warenwirtschaft abgeglichen werden. Der Shop stellt eine CSV-Schnittstelle bereit, die es ermöglicht, eine Artikelnummer mit einer anderen Artikelnummer zu matchen. Wie die Datei erstellt wird, weiß das E/D/E. Der Aufbau der Daten ist so einfach wie möglich gehalten: `EDEARTNR, EIGENE_NR`

Kundendaten

Gäste und neu registrierte Kunden werden über einen Standard-Debitor gehandlet. Wenn also keine Kundennummer im Shopsystem vorhanden ist, wird eine Standardnummer verwendet. Diese kriegt als Responses immer den Standard-Katalogpreis (oder den Preis, den der Shopbetreiber für sinnvoll hält) zurück. **Sobald die Live-Anbindung scharf geschaltet wird, ist die komplette Preisfindung aus dem Shop aufgehoben!**

Es besteht die Möglichkeit, seitens des Shops per Cronjob eine Datei per FTP zu laden. Dieser Cron synced die Kundendaten alle 15 Minuten und trägt in der Warenwirtschaft neu hinterlegte Kunden neu im Shop ein bzw. pflegt vorhandene Kundendaten nach. Falls also Lieschen Müller als Kunde im Shop hinterlegt ist, hier aber noch keine Kundennummer hat, kann das CSV-Modul die Kundennummer automatisch nachtragen. Wichtig ist hier der Primärschlüssel LOGIN, das dazu dient, den Kunden in

beiden Systemen zu erkennen.

```
LOGIN, PASSWORD, KUNDENNUMMER,VORNAME, NACHNAME, RECHNUNGSADRESSE
(Strasse,Firma, plz, ort)
```

Das Password kann leer gelassen werden, dann wird es ignoriert. Wenn es nicht leer gelassen wird, dann wird es im Shop per MD5-Hash verschlüsselt.