



SAP® R/3 und der Open Text Delivery Server

In der Praxis

31. Open Text Web Solutions Anwendertagung
Mannheim, 22. Januar 2013
Ralf Friedrichs

Übersicht

- **Einführung / Vorbetrachtung**
Was wollen wir tun ?
- **Die technische Seite**
Wie setze ich das jetzt um ?
- **Der Nutzen**
Warum sollte man völlig fremde Welten verbinden ?
- **Beispiel**
Ein einfaches Beispiel aus der Praxis



Einführung / Vorbetrachtung

Was wollen wir tun ?

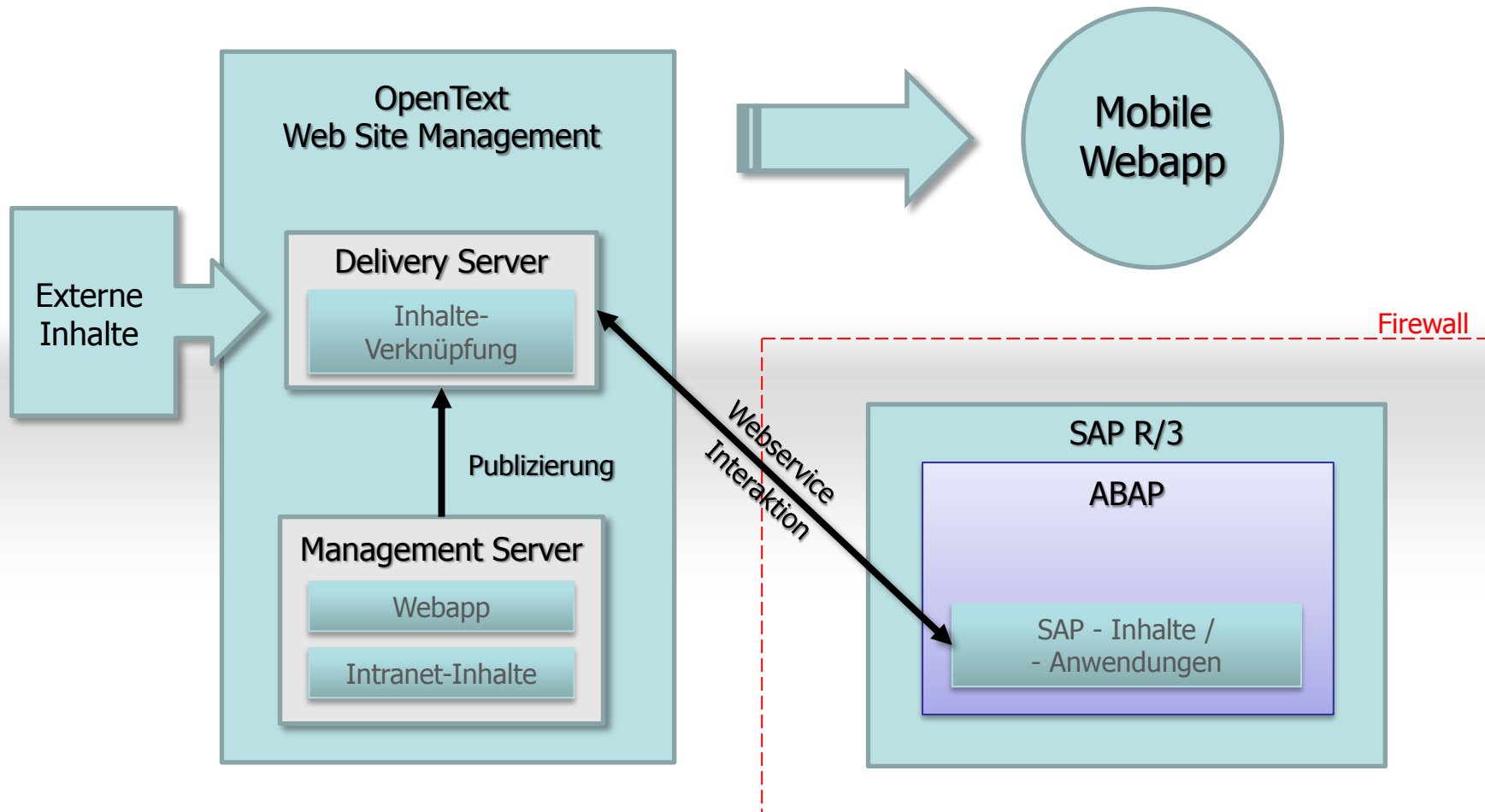
Einführung / Vorbetrachtung

- Es soll ein einfacher Zugriff auf bestimmte SAP-Anwendungen mittels mobilem Endgerät ermöglicht werden
- Der Zugriff soll nur auf das SAP R/3 System erfolgen - ohne SAP Netweaver Portal
- Die eingesetzten Kommunikationstechniken sollen im SAP R/3 System auf SAP-Standardfunktionalitäten basieren
- Der Visualisierung / die Oberfläche soll auf jQuery mobile / HTML 5 basieren
- Der Delivery Server soll die Oberfläche ausliefern und sie mit Daten versorgen
- Der Delivery Server steuert alle Verbindung zum SAP R/3 System

Einführung / Vorbetrachtung

- Die Webapp für mobile Endgeräte soll Einstiegspunkt für Inhalte aus unterschiedliche Quellen sein
- Neben SAP R/3 Inhalten und Anwendungen sollen mittels Delivery Server auch andere Inhalte angeboten werden
 - Intranet-Inhalte
 - Sharepoint-Inhalte
 - Nachrichten, Wetterinfos
 - etc.
- Es soll nicht transparent sein, welche Inhalte aus welchem System kommen

Einführung / Vorbetrachtung

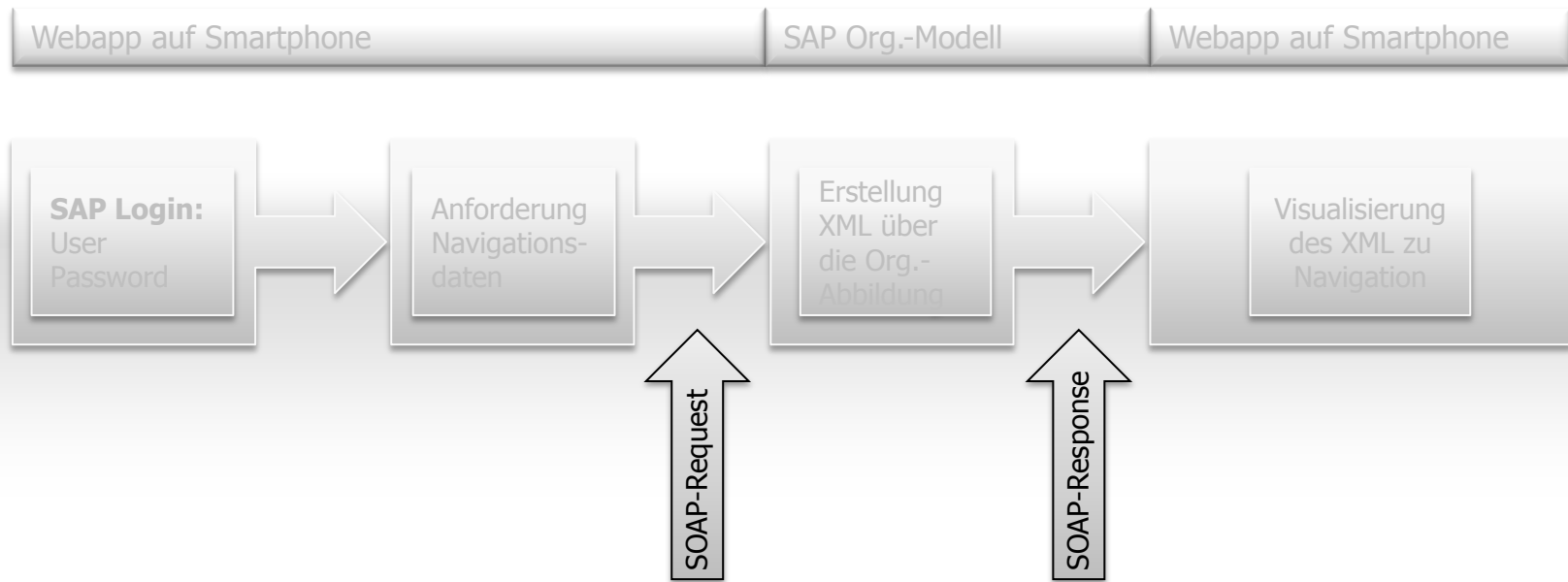


Die technische Seite

Wie setze ich das jetzt um ?

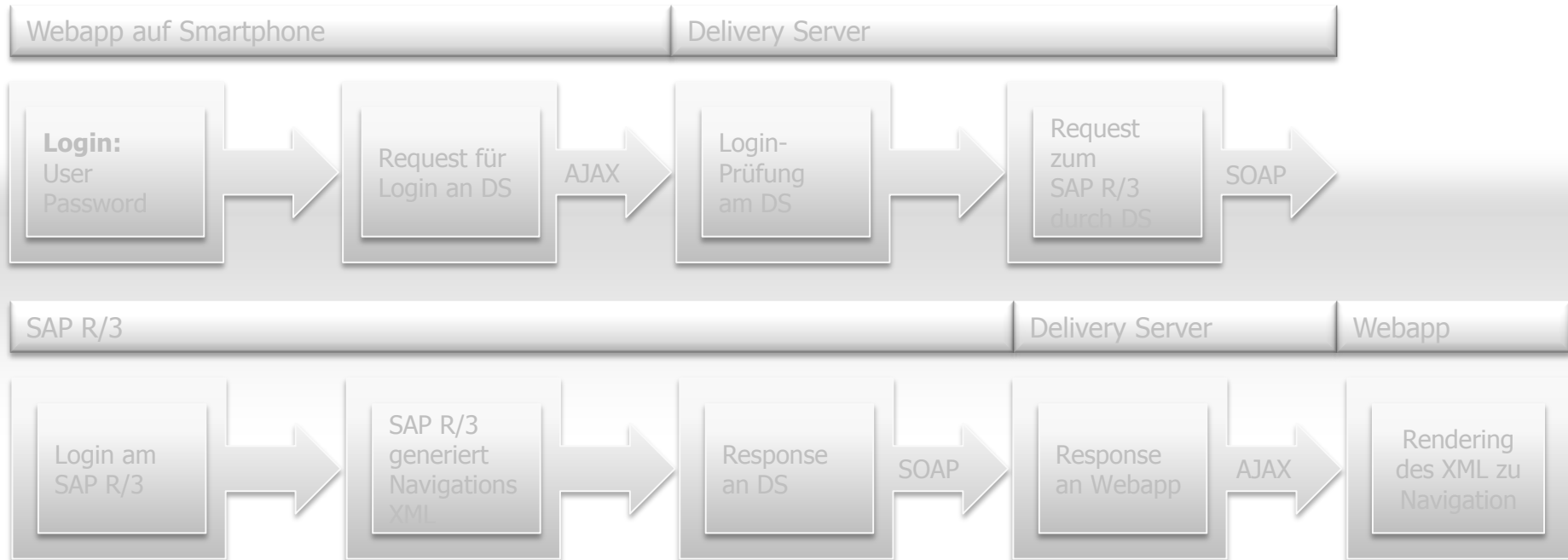
Die technische Seite – Delivery Server

Kommunikationsablauf zur Erstellung der SAP R/3 basierenden Navigation



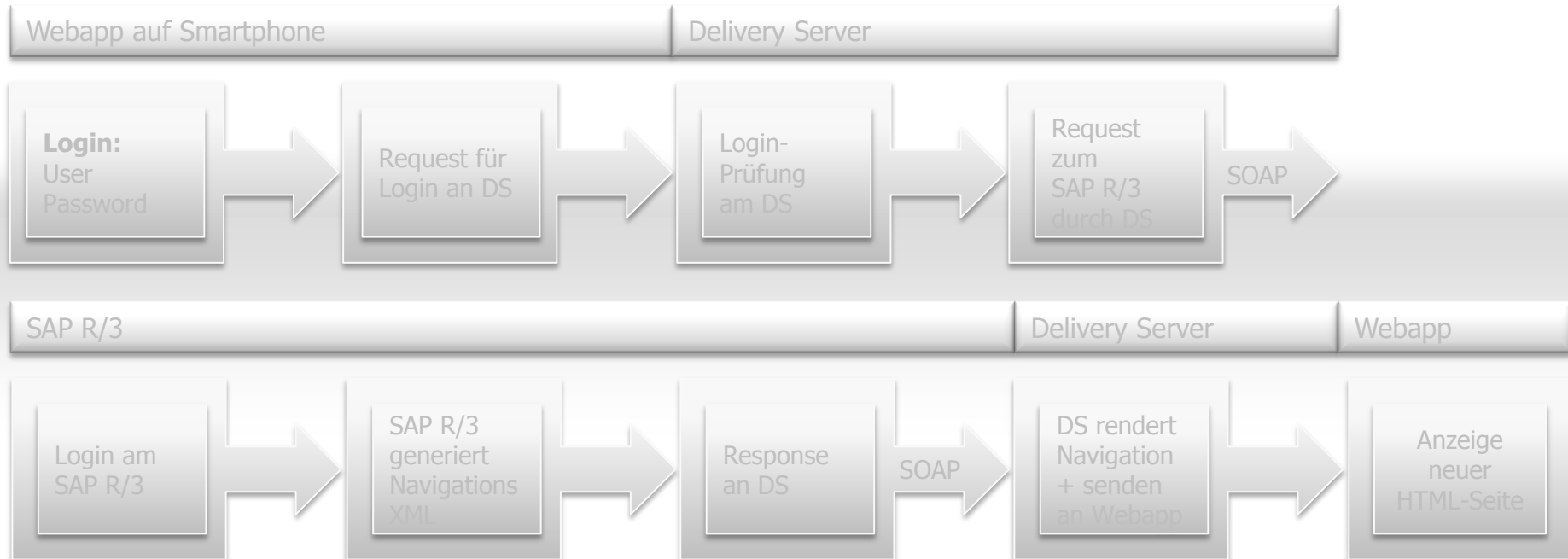
Die technische Seite – Delivery Server

Kommunikationsablauf zur Erstellung der SAP R/3 basierenden Navigation der Webapp mit Hilfe des Delivery Servers mittels Ajax



Die technische Seite – Delivery Server

Kommunikationsablauf zur Erstellung der SAP R/3 basierenden Navigation der Webapp mit Hilfe des Delivery Servers mittels HTML-Seite



Die technische Seite – Vorbereitung SAP R/3

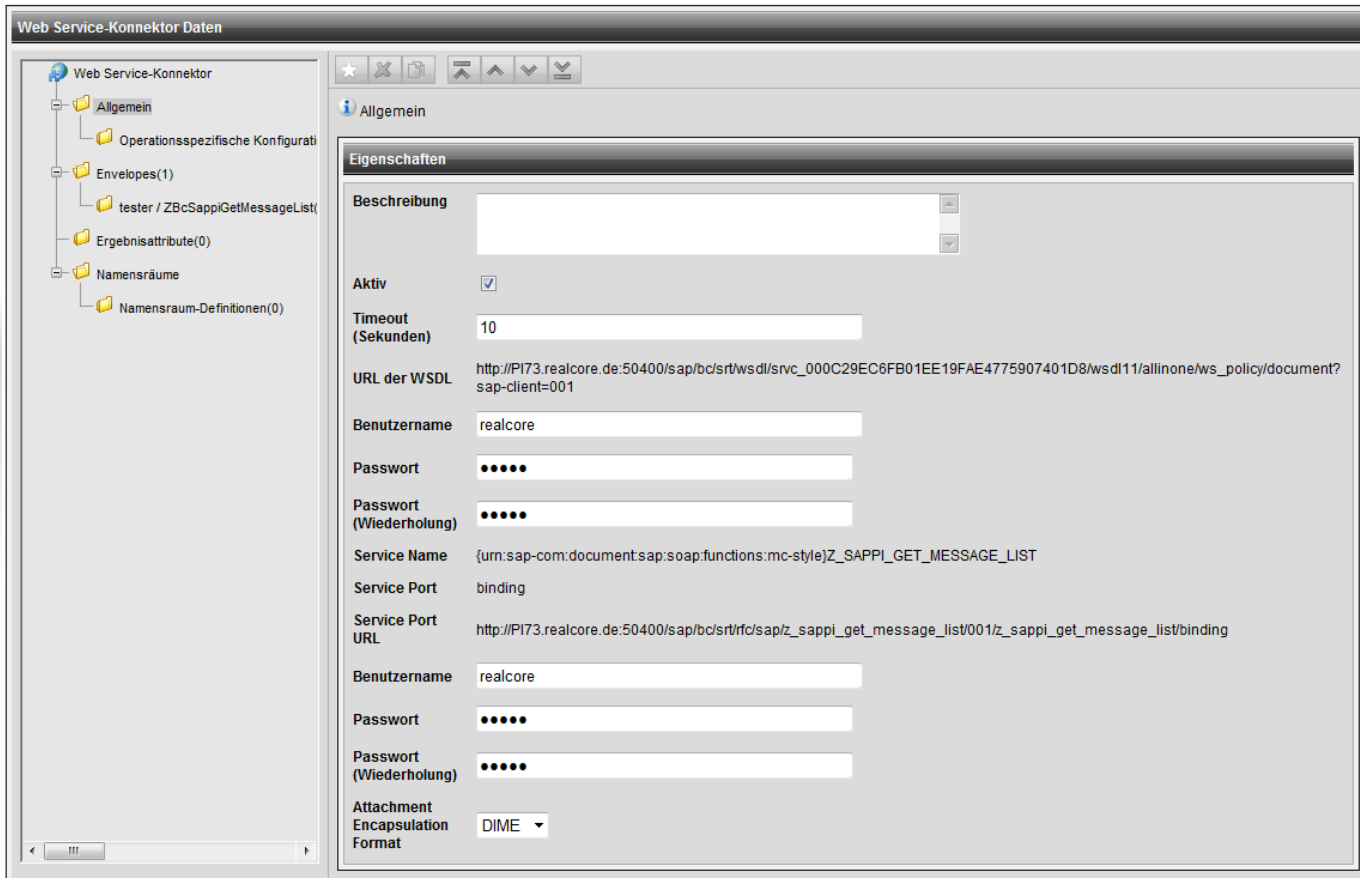
- Alle Benutzer, welche die Anwendung via Webservice nutzen wollten, müssen dem R/3 System bekannt sein und entsprechende Rollen besitzen.
- Es müssen Funktionsbausteine eingespielt werden, welche anhand der Nutzerberechtigung ein Navigations-XML mit den für die Webapp vorgesehenen Anwendungen bereitstellt.
- Für den Informationsaustausch müssen remotefähige Funktionsbausteine für die jeweilige Anwendung erstellt werden.
- Zu diesen Funktionsbausteinen müssen Webservices generiert werden. Die benötigten WSDL-Dateien werden automatisch vom SAP R/3 erstellt.

Die technische Seite – Delivery Server

- Vorbereitung der jQueryMobile basierende Oberfläche im CMS und Publikation zum Delivery Server.
- Konfiguration des Webservice-Konnektors (Lizenz notwendig).
- Daten für Webservice zusammenstellen und Dynament erstellen.
- Visualisierung der Webservice-Antwort (mittels JavaScript oder via XSLT)

Die technische Seite – Delivery Server

Einrichten / Konfiguration des Webservice-Konnektors



Web Service-Konnektor Daten

Allgemein

Eigenschaften

Beschreibung

Aktiv

Timeout (Sekunden) 10

URL der WSDL `http://PI73.realcore.de:50400/sap/bc/srt/wsdl/srv_000C29EC6FB01EE19FAE4775907401D8/wsdl11/allinone/ws_policy/document?sap-client=001`

Benutzername realcore

Passwort ●●●●

Passwort (Wiederholung) ●●●●

Service Name {urn:sap-com:document:sap:soap:functions:mc-style}Z_SAPPL_GET_MESSAGE_LIST

Service Port binding

Service Port URL `http://PI73.realcore.de:50400/sap/bc/srt/rfc/sap/z_sappl_get_message_list/001/z_sappl_get_message_list/binding`

Benutzername realcore

Passwort ●●●●

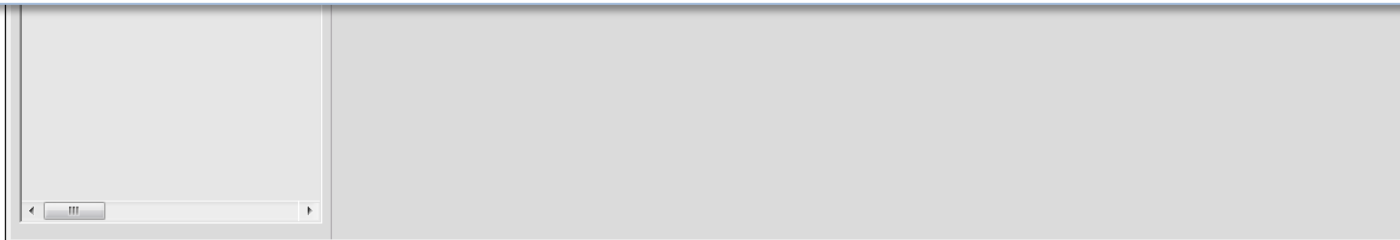
Passwort (Wiederholung) ●●●●

Attachment Encapsulation Format DIME

Die technische Seite – Delivery Server

Einrichten / Konfiguration des Webservice-Konnektors



```
<rde-dm:webservice name="og_mobile">
  <rde-rd:soap-message type="envelope">
    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:urn="urn:sap-com:document:sap:soap:functions:mc-style">
      <soapenv:Header/>
      <soapenv:Body>
        <urn:ZBcSappiGetMessageList>
          <ImFromDate>[#session:dateFrom#]</ImFromDate>
          <ImMessageCount>100</ImMessageCount>
          <ImSenderBs>BS_Niermann</ImSenderBs>
          <ImSenderIfname>SI_TESTDATEN_OUT</ImSenderIfname>
          <ImSenderIfns></ImSenderIfns>
          <ImSenderParty></ImSenderParty>
          <ImToDate>[#session:dateTo#]</ImToDate>
        </urn:ZBcSappiGetMessageList>
      </soapenv:Body>
    </soapenv:Envelope>
  </rde-rd:soap-message>
</rde-dm:webservice>
```



Der Nutzen

Warum sollte man völlig fremde Welten verbinden ?

Der Nutzen - Basisgedanken

- Es besteht ein Intranet auf Basis Open Text Websolutions
- Es besteht eine SAP-Infrastruktur mit diversen R/3 Anwendungen, aber kein SAP Portal verfügbar
-  Klassisch: Intranet und SAP R/3-Anwendungen sollen in einem Kontext für den Nutzer verfügbar sein.
-  Innovativ: Das Intranet und einige SAP R/3-Anwendungen sollten für mobile Endgeräte verfügbar gemacht werden.

Der Nutzen

- Es kann die bestehende Infrastruktur ohne Erweiterung genutzt werden
 - Keine zusätzlichen Lizenzen
 - Keine zusätzliche Hardware
- Es wird eine Webapp auf aktuellen Standard erzeugt (jQuery, HTML 5)
- Die Applikation zur Navigationserstellung wird durch das Standard-Transport-Feature von SAP eingespielt
- Die SAP R/3-spezifische Navigation wird vollständig automatisiert im SAP-Standard erstellt
- Lose Koppelung zwischen Webapp und SAP R/3

Der Nutzen

- Nutzung der CMS-Features zur Erstellung der Webapp-Basis
- Durch Einsatz des Delivery Servers können diverse Content- und Datenquellen in die Webapp eingemischt werden.
- Content-Personalisierung durch den Delivery Server möglich
- Responsive Webdesign möglich. Der Delivery Server kann Inhalte gezielt rendern.
- Das SAP-System ist für den Nutzer nicht transparent. Der Delivery Server für die komplette Kommunikation.

Der Nutzen - Einschränkungen

- Eine Open Text Websolutions Installation wird nicht speziell für diesen Einsatzzweck angeschafft
- Voraussichtlich Anpassungen bzgl. Firewall- und Zugriffsschutz-Richtlinien
- Named-User Prinzip im SAP R/3 System
- Oberflächen für Applikationen müssen erstellt werden
- Anwendungen müssen Webservice-fähig gemacht werden

Beispiel

Ein einfaches Beispiel aus der Praxis

Beispiel – Passwort Self Service



Beispiel – Passwort Self Service

PSS Mobile

Passwort Self Service - Anfrage:

Benutzer-ID

SAP-System

Mandant

E-Mail Adresse

[← Abbrechen](#) [Ausführen →](#)

PSS Mobile

Passwort Self Service - Anfrage:

Benutzer-ID

SAP-System

Mandant

E-Mail Adresse

TAN

[← Abbrechen](#) [Ausführen →](#)

PSS Mobile

Ihr Token wurde erfolgreich bestätigt.
 Bitte vergeben Sie jetzt Ihr neues Passwort

Neues Passwort

Neues Passwort wiederholen

[← Abbrechen](#) [Ausführen →](#)



Vielen Dank für Ihr Interesse

Ralf Friedrichs

Unitleiter Websolutions

ralf.friedrichs@realcore.de

+49 (201) 178 968 0