



White Paper

Zukunftsprojekt

GEVITAS WebService

© GEVITAS GMBH

Dieses White Paper beschäftigt sich mit GEVITAS-WebService, einem Zukunftsprojekt der GEVITAS GmbH. Es sammelt Ideen, Visionen und Gedanken zum Thema WebService.

Was ist ein WebService?

Ein Webservice ist eine Software auf einem Server im Intranet oder Internet. Er wartet auf Kommandos von außen, liefert die Ergebnisse von (Datenbank-)Abfragen oder Berechnungen an den Client zurück. Das Internet 2.0 ist voll von WebServices! Wenn man auf seinem PC-Desktop ein Wetter-Gadget hat, so bedient sich diese mit ziemlicher Sicherheit eines WebService. Auch das Smartphone, das die neuesten Fußball-Ergebnisse zeigt, greift auf einen Webservice zu.

Webservices sind nicht Neues! Schon vor vielen Jahren haben sich die großen Software-Hersteller Gedanken über eine Normierung der Schnittstellen und der Kommunikation gemacht und Standards entwickelt (COM, CORBA usw.). Vieles hat die Erwartungen erfüllt, manches nicht.

Eine aktuelle Norm (ein Protokoll) für den einfachen Datenaustausch ist z.B. „SOAP“ (Simple Object Access Protocol). Es wird im Allgemeinen mit dem Internet-Protokoll http verwendet, um Daten im sog. „XML-Format“ auszutauschen.

Sinn der Sache ist, dass zukünftige Systeme mithilfe der Bausteine aus dem Web Daten und Dienste bereitstellen, ganz gleich, ob aus einem Web-Browser heraus, in einer Windows-Software oder einer mobilen App.

Was hat das mit GEVITAS zu tun?

Wir wollen zukünftig keine Wetter-App mit zugehörigem Webservice anbieten (obwohl: man kann ja nie wissen...)!

Die Vision ist vielmehr, dass zukünftige GEVITAS-Anwendungen (egal, auf welcher Hardware, mit welchem Betriebssystem und mit welcher Programmiersprache) nur die Oberfläche beinhalten. Die eigentliche „Intelligenz“ sitzt aber im Internet oder Intranet: Ein Webservice!

GEVITAS-Software oder –Apps sind in der Regel Datenbank-Anwendungen, d.h. sie zeigen Informationen aus der Datenbank an, verarbeiten und speichern Daten. Die Art der Verarbeitung unterliegt dabei bestimmten Regeln, z.B. bei der Dispo-Abfrage oder dem Verbuchen einer Lagerausgabe. Es bietet sich also an, die Regeln der Datenverarbeitung (Logik) an einer zentralen Stelle zu hinterlegen und diese zentrale Stelle über festgelegte Kommandos zu steuern.

Entfällt REFLEX oder GEVAS-Professional damit?

Nein. Diese Programme wird es weiterhin geben. Irgendwann werden sie halt von einer ganz neuen Generation abgelöst, die eben den Web-Service benutzt.

Werden REFLEX oder GEVAS-Professional nicht mehr weitergepflegt?

Natürlich werden wir unsere Flaggschiffe **noch über Jahre hinweg** weiterpflegen! Wir reden hier in diesem White Paper über eine Entwicklung, die sich über Jahre hinziehen wird. Die Roadmap zu diesem Projekt sieht z.B. vor, dass Ende des Jahres 2012 festgelegt wird, mit welchem Protokoll (z.B. „SOAP“ oder „REST“) der Webservice betrieben wird. Dazu sind umfangreiche Untersuchungen notwendig.

Die Überlegung ist, im Jahre 2013 einen Service für mobile Scanner zu erstellen und erst einmal praktische Erfahrungen damit zu sammeln. Dabei haben wir den Vorteil, dass MobiFlex in .Net/C# geschrieben ist und viele Funktionen (z.B. Lagerausgabe/-Rücknahme) schon vorhanden und ausgereift sind. Diese könnte man dann relativ einfach auf den Webservice übertragen.

Client/Server-Anwendungen wie REFLEX oder GEVAS-Professional haben auch weiterhin ihre Daseins-Berechtigung!

Entfällt die BDE damit?

Ja. Mit dem Webservice ist die BDE Vergangenheit! Der Web-Service benutzt ADO.Net als SQL-Schnittstelle.

Ein aktuelles Beispiel

Einige Barcode-Scanner der neuesten Generation kommen mit dem Android®-Betriebssystem. Das bietet sich im Prinzip an, schließlich ist Android® ein modernes und verbreitetes System. Ein Problem ergibt sich (für uns) aber daraus, dass Android® normalerweise keine Möglichkeit bietet, mit SQL-Servern zu kommunizieren. Es gibt keinen offiziellen Weg, SQL-Server wie Microsoft SQL-Server®, MySQL® oder Oracle® anzusprechen. Der Grund liegt wohl darin, dass in jeder Android-Maschine ein SQL-Server eingebaut ist („SQLite“), wie übrigens auch in Apples® iOS. Zwar gibt es verschlungene Wege, über JDBC auf dem einen oder anderen Weg zu SQL-Servern zu kommen, aber es fragt sich, ob man diesen Weg gehen sollte.

Eine Lösung des Problems ist der GEVITAS-Webservice:



<p>Der Client scannt einen Barcode. Nun muss geprüft werden, ob es eine gültige Nummer ist, ob der Artikel ausgegeben werden kann, in Reparatur ist usw.</p> <p>Das schickt er eine Anfrage an den GEVITAS-WebService: <i>„Ist die Inventarnummer 4712 korrekt und darf dieser Artikel ausgegeben werden?“</i> (natürlich nicht in diesen Worten...)</p>	<p>Der Webservice hat viele Funktionen. Eine davon heißt <i>„Ist die Inventarnummer X korrekt und darf dieser Artikel ausgegeben werden?“</i></p> <p>Diese Funktion erzeugt nun entsprechende SQL-Kommandos und schickt Abfragen zum Datenbank-Server. Diese werden analysiert und eine entsprechende Antwort an den Client zurückgeschickt.</p>	<p>Der SQL-Server steht im Intranet in Verbindung mit dem Webservice*).</p>
<p>Der Client erhält die Antwort und zeigt z.B. die Fehlermeldung „Dieser Artikel ist derzeit in Reparatur“ an.</p> <p>Der Client kann dabei über das Intranet (W-LAN) oder über das Internet verbunden sein.</p> <p>Die Kommunikation erfolgt über HTTPS/XML.</p>	<p>Der Webservice liegt auf einem Server, der nach außen mit dem Intranet oder Internet verbunden ist. Nach innen ist er über das lokale Netzwerk (LAN) mit dem SQL-Server verbunden*).</p>	<p>Der SQL-Server ist nur mit dem Intranet (LAN) verbunden*).</p>

*) Natürlich kann der Webservice auch direkt auf der SQL-Server-Maschine laufen. Sicherer und schneller ist aber die oben gezeigte Konstellation.

Wenn man diesen Gedanken weiterverfolgt, stellt sich die Frage: Warum sollte man dieses System nur mit mobilen Barcode-Scannern betreiben? Warum nicht auch in der fernen Zukunft mit REFLEX, GEVAS-Professional usw.?

- ❖ Wahrscheinlich ist der GEVITAS-WebService nicht eine einzige Software sondern besteht aus einer Reihe von spezialisierten Programmen, z.B. ein Lager-Service, ein Stammdaten-Service, ein Dispo-Service usw.

Vorteile des Webservice

- **Zukunftssicher:** Der Webservice kann von jeder Hardware und jedem Betriebssystem verwendet werden. Oder von einem Web-Browser. Mit dem Webservice können wir dem „Krieg der Betriebssysteme“ gelassen entgegen sehen.
- **Schnell (?)**: Weil nur die relevanten Daten über das Netz müssen, ist die Netzwerk-Auslastung überschaubar, auch relativ langsame Verbindungen sind kein Problem. Siehe auch Nachteil!
- **Mobil:** Daher kann man von überall in der Welt, wo es Internet gibt, auf sein REFLEX oder GEVAS-Professional zugreifen!
- **Zentral:** Die Programm-Logik liegt zentral in einer Software: Dem GEVITAS-WebService. Das vereinfacht und zentralisiert die Software-Wartung.

Nachteile des Webservice

- **Ausfall:** Fällt der Webservice oder das Netzwerk aus, kann niemand mit der Software arbeiten. Allerdings gilt das auch schon jetzt für den SQL-Server.
- **Hardware-Anforderung an den Server:** Der Server, auf dem der Webservice läuft, muss reichlich mit Power und Speicher ausgerüstet sein.
- **Sicherheit:** Wenn der Webservice am Internet hängt, muss die Sicherheit gewährleistet werden. Dazu gehört:
 - HTTPS-Verbindung
 - Authentifizierung am Webservice
 - Spezielles Recht für den SQL-Server
 - Zugriffssteuerung der REFLEX/GEVAS-Professional-Rechte
- **Datenvolumen:** Alle XML-basierenden Protokolle wie SOAP erzeugen recht große Dokumente für den Datenaustausch. Es muss untersucht werden, inwieweit die Performance bei umfangreichen Daten ausreicht. Sollte sich im Laufe der Vor-Untersuchungen herausstellen, dass das Datenvolumen wegen SOAP zu groß und damit zu langsam ist, muss die Alternative REST („REpresentational State Transfer Architektur“) geprüft werden.

Technik – Das SOAP-Protokoll

SOAP ist ein Industriestandard, der den plattformübergreifenden Datenaustausch ermöglicht. Er basiert auf dem Standard-Protokoll HTML und dem Datenformat XML. Einzelheiten siehe [Wikipedia](#).

SOAP wird von .Net-Framework (Microsoft) sehr gut unterstützt. In der JAVA-Welt ist JSON verbreiteter, aber auch SOAP wird unterstützt.

Technik – Das REST-Protokoll

Auch REST ist ein Industriestandard, der auf dem Standard-Protokoll HTML basiert. Wenn REST mit dem Datenformat XML benutzt wird, erzeugt es genau so viel Datentransfer wie SOAP.

Allerdings kann man REST auch mit JSON („JavaScript Object Notation“) verwenden. JSON ist viel effektiver als XML, allerdings nicht so weit verbreitet.

Die Syntax von JSON ist kompakt. Die Beschreibung einer Person würde etwa so aussehen:

```
{
  "Vorname": "Harry ",
  "Nachname": "Hastig",
}
```

In XML würde das so aussehen (ohne Attribute):

```
<Person>
  <Vorname>Harry</Vorname>
  <Nachname>Hastig</Nachname>
</Person>
```

XML erzeugt also größere Daten als JSON. Ob das relevant ist, müssen Untersuchungen zeigen!

Technik - .Net-Framework oder JAVA?

Derzeit ist geplant, den GEVITAS-WebService auf Basis des .Net-Frameworks zu entwickeln. Die Gründe:

- SOAP ist Bestandteil vom .Net-Framework von Microsoft aber auch in JAVA einfach zu handhaben.
- MobiFlex ist in .Net/C# geschrieben und beinhaltet schon eine Vielzahl an Funktionen, die der Webservice können muss. Diese könnte man relativ einfach portieren, weil MobiFlex jetzt schon modular aufgebaut ist und die sog. „[Drei-Schichten-Architektur](#)“ relativ gut beinhaltet. Diese Architektur trennt eine Software in drei Schichten:
 - o Präsentation, also die Eingabemasken, Listen usw.
 - o Anwendungslogik
 - o Speicherung (SQL-Funktionen)