

SOA mit der VCL einsetzen

Vom RPC, Remote Procedure Call 1985, IPC (Inter Process Communication) zu DCOM und **Web Service** 2001 hin zur SOA, Service Oriented Architecture 2006

<http://max.kleiner.com>



Agenda

- Was ist SOA und was bietet die VCL /CLX.
- Kernidee (Beispiel) ist, gesammelte Daten aus n-Clients mit einer Applikation oder einem Dienst via SOAP zentral an einen Server zu senden.
- Die Umsetzung zeigt mit WebServices ein Framework zur entfernten Speicherung von Daten (Storage as Service) mit Hilfe eines Assistenten (Service Orchestration).
- Warum SOA ein Konzept und keine Technik ist!

What's SOA?

- The term SOA (Service Oriented Architecture) expresses a software concept that defines the use of services to support the requirements (use cases) of users.
- Unlike traditional point-to-point architectures, SOAs comprise loosely coupled, reusable and highly interoperable application services.

wikipedia.org 20. March 06

A Service ?

- (Ideally) a self-contained, stateless business function that accepts requests and returns something through a well defined, standard interface over a communication infrastructure (service-bus).
- Most SOAs are based on WebServices (SOAP over HTTP) which is a synchronous request/response messaging interaction.
- Be careful: loose coupling sounds simpler but's harder to implement and increases administration. While agility accepts that some requirements are never stable we may consider SOA as a bridge between the gap that systems are never unified!

kleiner.com 06

SOA Patterns

Communi- cation	Design Pattern	Platform Independend	Loose Coupling
sync	Abstract Factory, Prototype	Mittel	Ja (WSDL)
sync	Bridge, Observer, MVC	Mittel Hoch	Mittel
sync/async	Chain of Responsibility	Mittel	Ja (local or WAN)
sync	Command	Sehr hoch	Mittel
sync/async	Facade	Sehr hoch	Sehr hoch
sync/async	Mediator	Sehr hoch	Sehr hoch

VCL Ways

Client	Server	Format	Protokoll	Performanz	Internet-tauglich?
Win/Lin	Internet Server	XML, HTML	SOAP, HTTP	Mittel	Ja
Win/Lin .net	Applikation Server	Binär	TCP, UDP	Hoch	Mittel
Beliebig	Middleware	XML-RPC	Corba, HTTP	Mittel	Ja
Win/Lin	DWS	Stream	Sockets	Hoch	Mittel
Win32	.net	NDR	DCOM	Hoch	Nein
.net	.net	NDR	Remoting	Hoch	Mittel

NDR: Network Data Representation

SOA in Com or .net ?

COM-Space	.net-Space
AddRef/Release	Garbage Collector
Binäre Typbibliothek (IDL), Resource	Metadaten in der Assembly
HRESULT-Rückgabewert	Exception Handling
Apartment (STA)	Free-Threading
Automation-Datentypen	.net-Datentypen

VCL Technologies

- Typische Erscheinungsformen
 - Client mit Browser-GUI oder Application Loader
 - Client als Thin/Rich-Anwendung
 - Protokolle TCP/IP, HTTP (SOAP), DCOM, RMI
 - reine MDA-Clients mit ECO III
 - Win Services (local or pipe) mit Unit ServiceManager
 - gemixte .net- und DCOM-Clients
- Wie betrachtet man die Typisierung?
 - Daten (Parameter/DataSet) oder Objekte?

SOA Beispiel

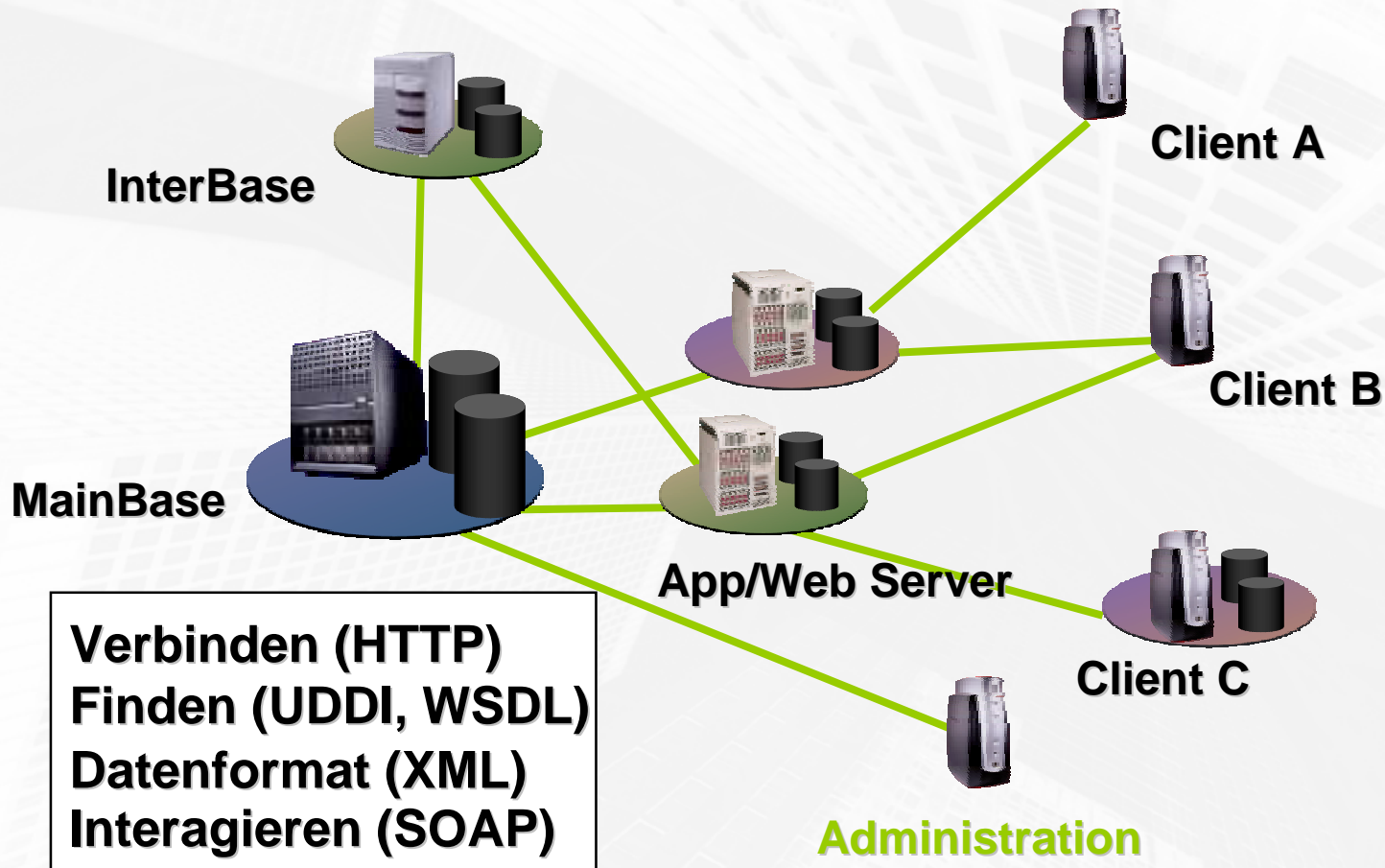
- Ein Netz von mobilen Rechnern sendet periodisch Daten via Service Manager an einen zentralen Server, der die ausgewerteten Daten dann per Browser für die Benutzer bereithält.
- In der folgenden Fallstudie möchte ich die Grundzüge dieser Technik näher bringen um SOA gerecht zu werden.



Worum geht es ?

- VCLScanner.exe als Client (Compilat)
- VCLScannerServer.exe als Webservice
- Apache WebServer
- InterBase DB / DBExpress Connector
- Die Elemente des RDS:
 - WSDLPublish
 - Dispatcher
 - Invoker
 - DataSetProducer

SOA konkret



WebService Technik

- Dienste im Netz, die über Standard-Protokolle erreichbar und verwaltbar sind
- Datenaustausch basiert auf XML
- Plattform- und unternehmensübergreifende Transaktionen kombinierbar
- OO-Technik mit Interfaces realisiert
- HTTP-Port 80 in der Regel offen
- WebService ist EINE Technik zu SOA

Schlüsseltechnologie

Web
Service
Benutzer

Finden: UDDI

<http://FindAService.ch>

XML mit Link zur Web Site

Business
Repos.

Realisieren: WSDL

<http://adb2.ch/scanner.exe/WSDL>

XML mit Servicebeschreibung

Web
Service

Ausführen: SOAP

<http://adb2.ch/serverscanner.exe>

XML als Prozeduraufruf

Webdienst-Katalog UDDI

- Universal Description, Discovery and Integration
- „Gelbe Seiten“ des Internet für Webdienste
- Gremium aus ca. 120 Firmen für die Propagation von Webdiensten
- <http://www.xmethods.net> and click the BabelFish Web Service. Also notice the path to the WSDL file for the service:
<http://www.xmethods.net/sd/2001/BabelFishService.wsdl>.
Or to Google WebService API: www.google.com/apis/

SOAP Webmodul

- Die Komponente `THHTTPSoapDispatcher` empfängt und beantwortet hereinkommende SOAP-Nachrichten und entscheidet, welches Interface man verwendet.
- Das korrekte Objekt, welches das Interface realisiert, wird an den `PascalInvoker` weitergeleitet, der die richtig ausgepackte Methode mit den formatierten Argumenten einschliesslich Errorcodes in den eigentlichen OP-Aufruf inklusive Parametern transformiert.
- WSDL ist ein XML-Derivat zur Beschreibung der Schnittstellen von WebServices. Nachrichtenstrom-Formate und Funktionsaufrufe werden definiert.
- `HTMLPublish` publiziert **WebServiceDescriptionLanguage**

Das Web Modul

type

```
TWebModule1 = class(TWebModule)
```

```
WSDLHTMLPublish1: TWSDLHTMLPublish;
```

```
HTTPSoapDispatcher1: THTTPSoapDispatcher;
```

```
HTTPSoapPascalInvoker1: THTTPSoapPInvoker;
```

```
DataSetTableProducer1: TDataSetTableProducer;
```

Das Interface

type

{ Invokable interfaces must derive from IInvokable }

IVCLScanner = interface(IInvokable)

['{8FFBAA56-B4C2-4A32-924D-B3D3DE2C4EFF}']

function PostData(const UserData: WideString; const
 Checksum: DWORD): Boolean; stdcall;

procedure PostUser(const Email, FirstName,
 LastName: WideString); stdcall;

end;

Die Implementation des Interface

```
TVCLScanner = class (TInvokableClass, IVCLScanner)
public
    function PostData(const UserData : WideString;
        const CheckSum : DWORD): Boolean; stdcall;
    procedure PostUser(const Email, FirstName,
        LastName : WideString); stdcall;
end;
Implementation ...
```

Client Aufruf

var

WS: IVCLScanner;

- WS:= HTTPRIO1 as IVCLScanner;
- WS.PostData(reFinalResults.Text,CRC);

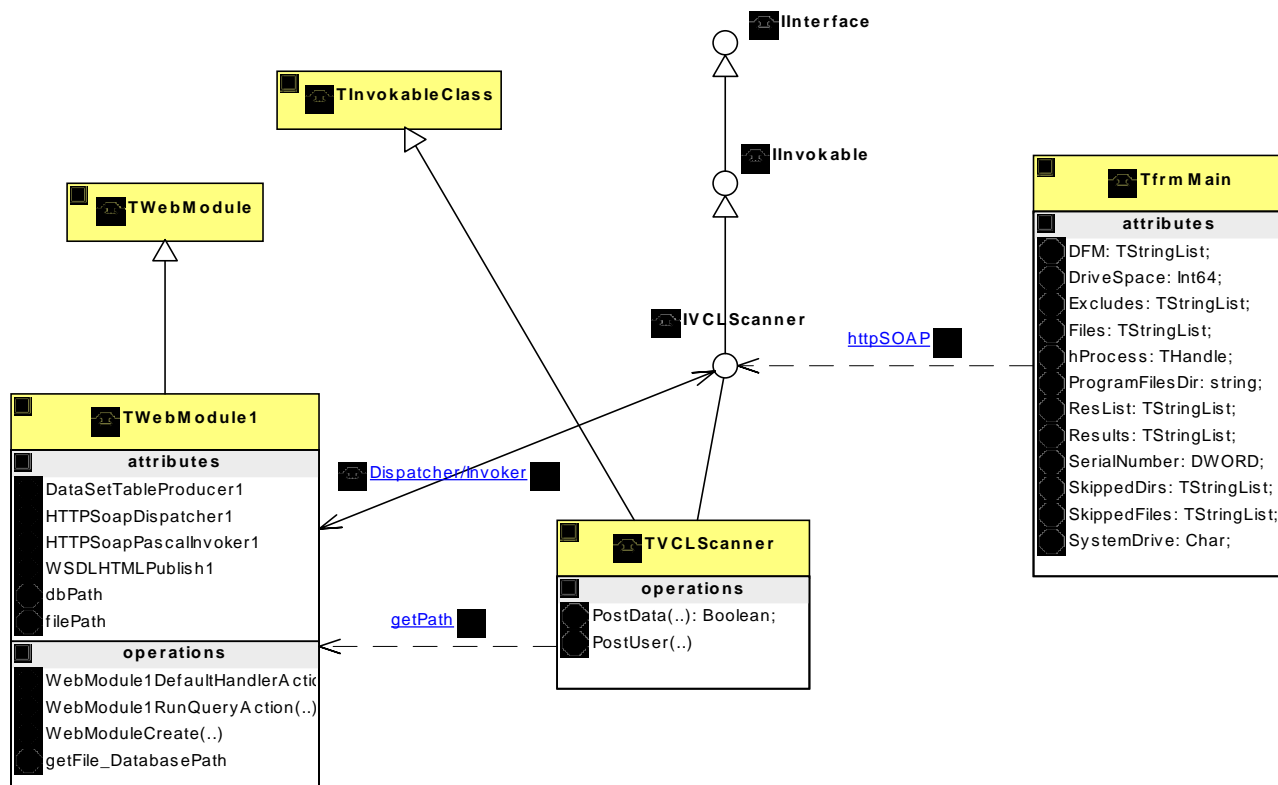
HTTPRIO1:URL {also getIVCLScanner() over WSDL Import}
<http://adb2/scripts/vclscannerserver.exe/soap/IVCLScanner>



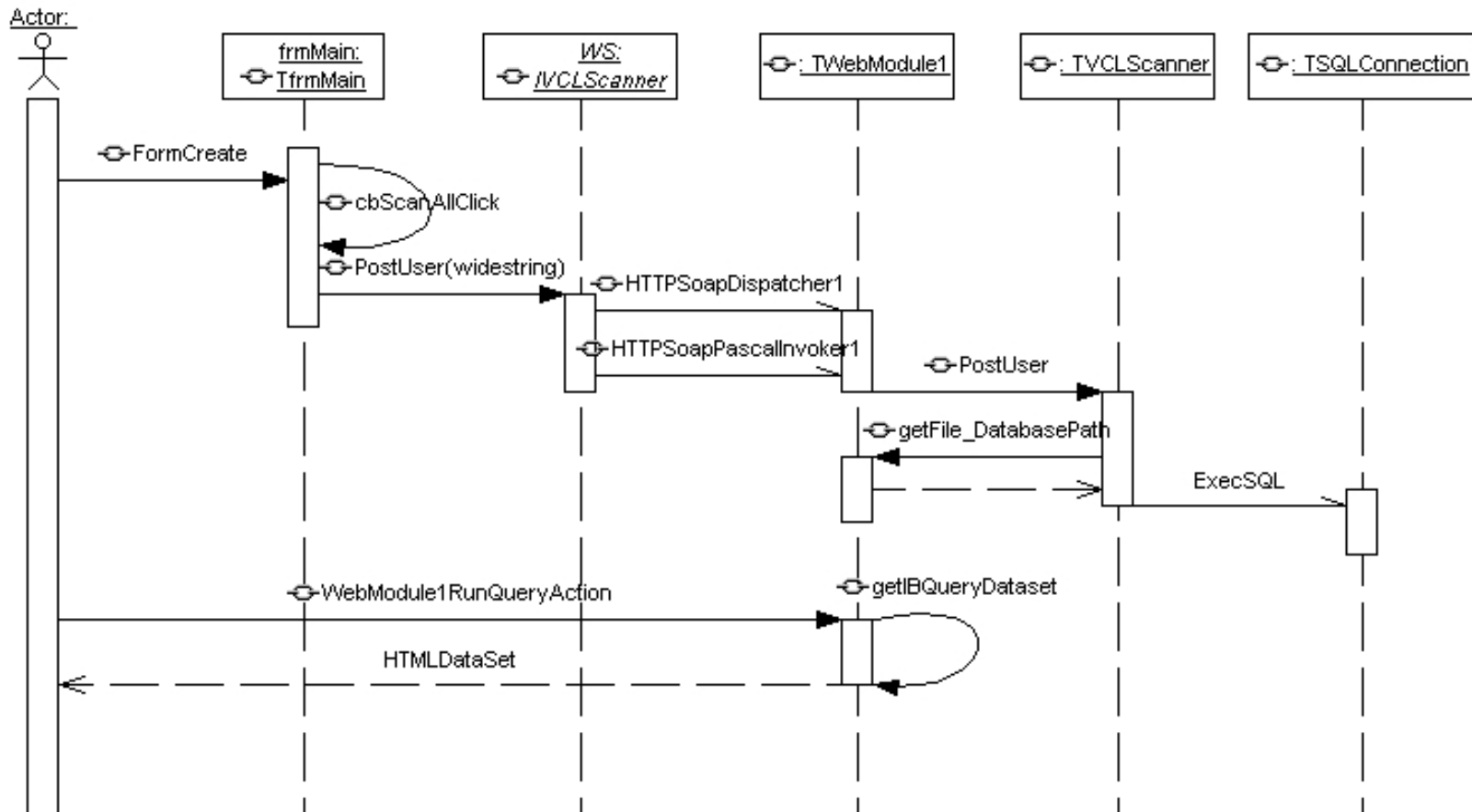
Die sieben DBExpress Schritte

1. `Connection:= TSQLConnection.Create(NIL);`
2. `with Connection do begin ...`
 `Params.Add()...`
3. `DataSet:= TSQLDataSet.Create(NIL);`
4. `with DataSet do begin ...`
5. `SQLConnection:= Connection;`
6. `CommandText:=`
 `Format('insert into ACL_LN values("%d", "%s", "%s",`
 `"%s", "%s")', [10, Email, FName, LName, ACLdate]);`
7. `ExecSQL;`

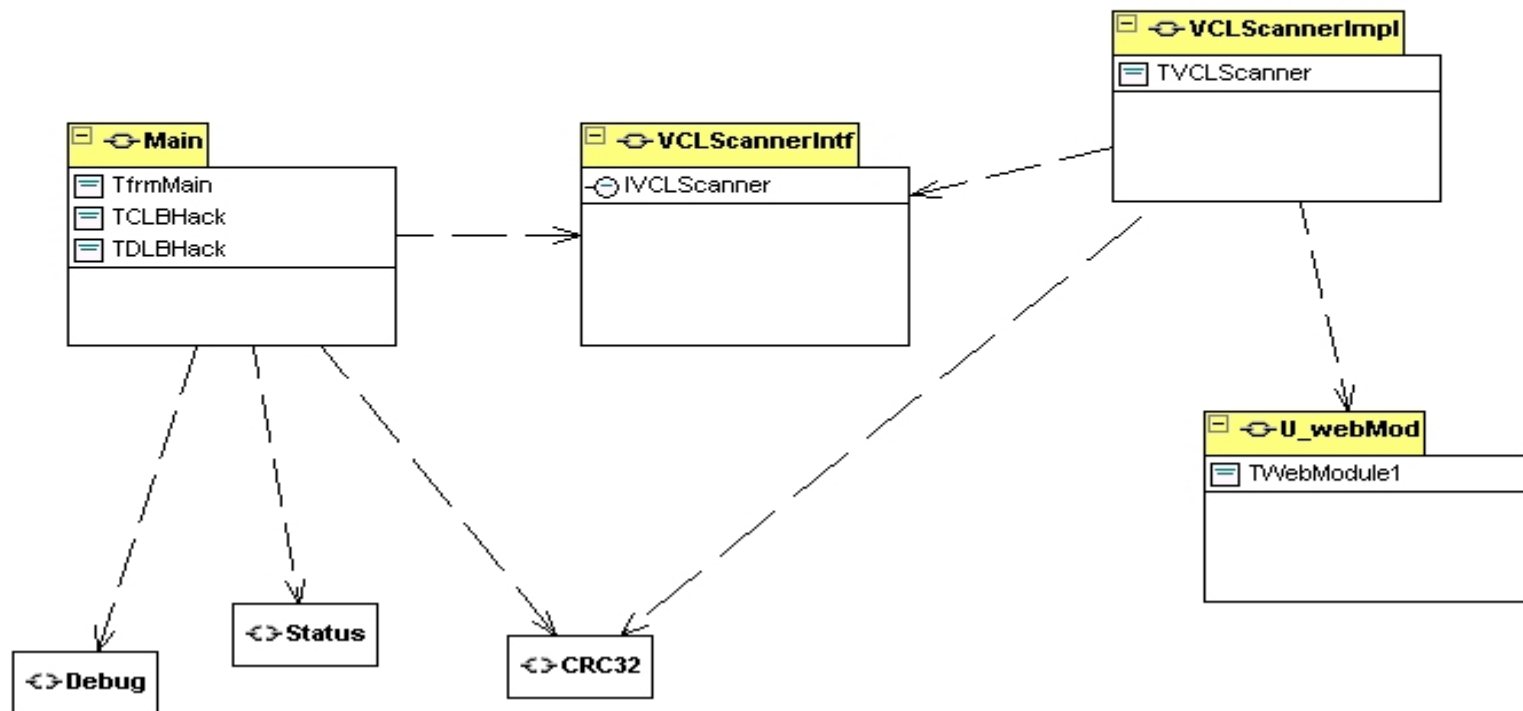
Struktur als Class Diagram



Das Sequenzdiagramm



Architektur



Erweitern des Interface

[IWSDLPublish]

IWSDLPublishPort=http://../soap/IWSDLPublish

[IVCLScanner]

IVCLScannerPort=http://../soap/IVCLScanner



WS Vorteil gegenüber Scripts

- OO-Technik mit Klassenbildung
- Direkt aus dem Code instanzierbar
 - Auch ohne Browser oder WAN möglich
 - Microsoft selbst betont den Willen zur Einbindung anderer Hersteller und Plattformen
 - Echte typensichere Compile erlauben eine stabile Nutzung mehrerer mobiler Plattformen und Sprachen
 - Dank des Verzeichnisdienstes immer up-to-date

Anspruch und Wirklichkeit

- Nicht jeder Service oder Prozess ist nützlich genug, verteilt, orchestriert und veröffentlicht und verwaltet zu werden.
- Zu viele nachrichtenorientierte Services erhöhen den Verwaltungsaufwand und vermindern Performance und Stabilität.
- SOA ist wenig typensicher und autonom, will heißen, der Hund ist zur Laufzeit begraben!

One last compile ?

„You have achieved perfection in design, not when you have nothing more to add, but when you have nothing more to take away“ .

Antoine de Saint Exupery

F1: Fragen und hoffentlich Antworten ?

Service Interaction Patterns:

www.eaipatterns.com

www.serviceinteraction.com

www.conversationpatterns.com

