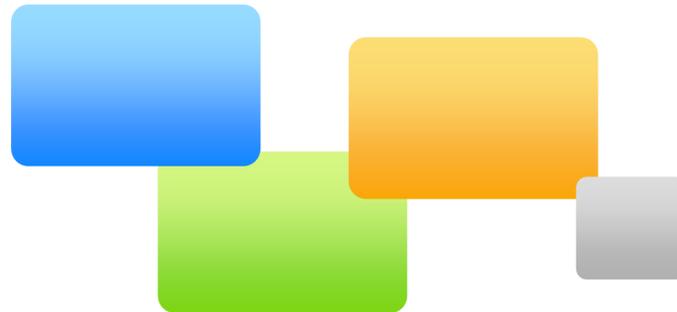


5. Web Development Treffen

Rich Internet Applications mit Adobe Flex und PHP/Java/.NET Backend



Björn Karpenstein

::: Geschlecht: männlich, geboren am 27/12/1980
de

::: Telefon: 0661 / 2069494

::: Email: bjoern.karpenstein (at) capri-soft.

Abschluss

Diplom-Informatiker (FH) 2006, Hochschule Fulda	2001-2006
Fachoberschule Bad Hersfeld, 2000, Europaschule Obersberg	1999-2000

Berufserfahrung

Application Manager, B. Braun Melsungen AG, Supply Chain Tools & Systems	04/2009-present
Software Engineer, B. Braun Melsungen AG, Architecture & Development	03/2006-03/2009
Delphi Entwickler, afs Software GmbH & Co. KG, Bad Hersfeld	studienbegleitend

Was ist
Adobe **Flex**?

Wie erstellt man eine
Flex **Anwendung**?

Was kann man
damit **machen**?

Signifikante
Merkmale

Was ist Adobe Flex?

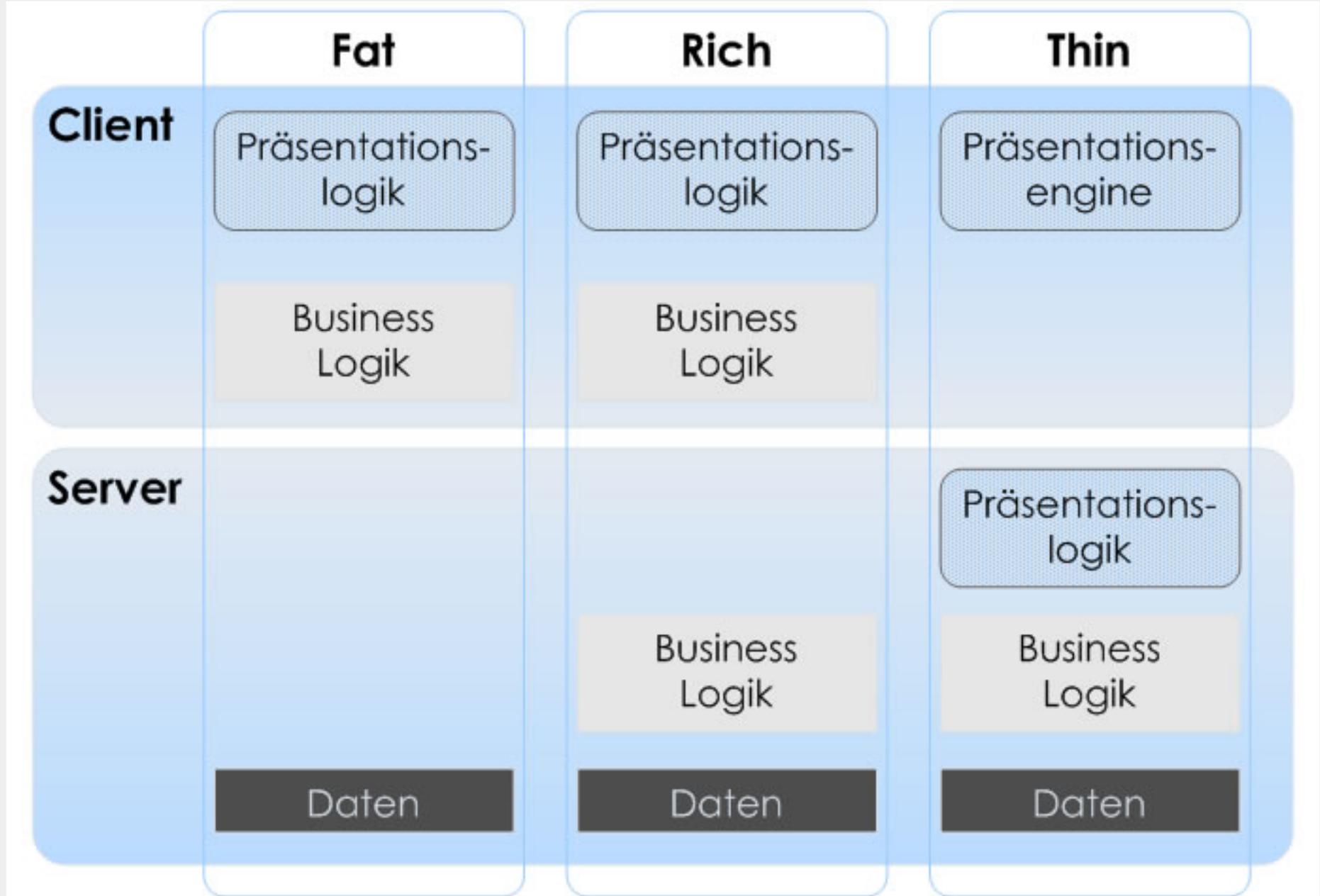
Framework/Architektur

Was kann man
damit machen?

Wie erstellt man eine
Flex Anwendung

Signifikante
Merkmale

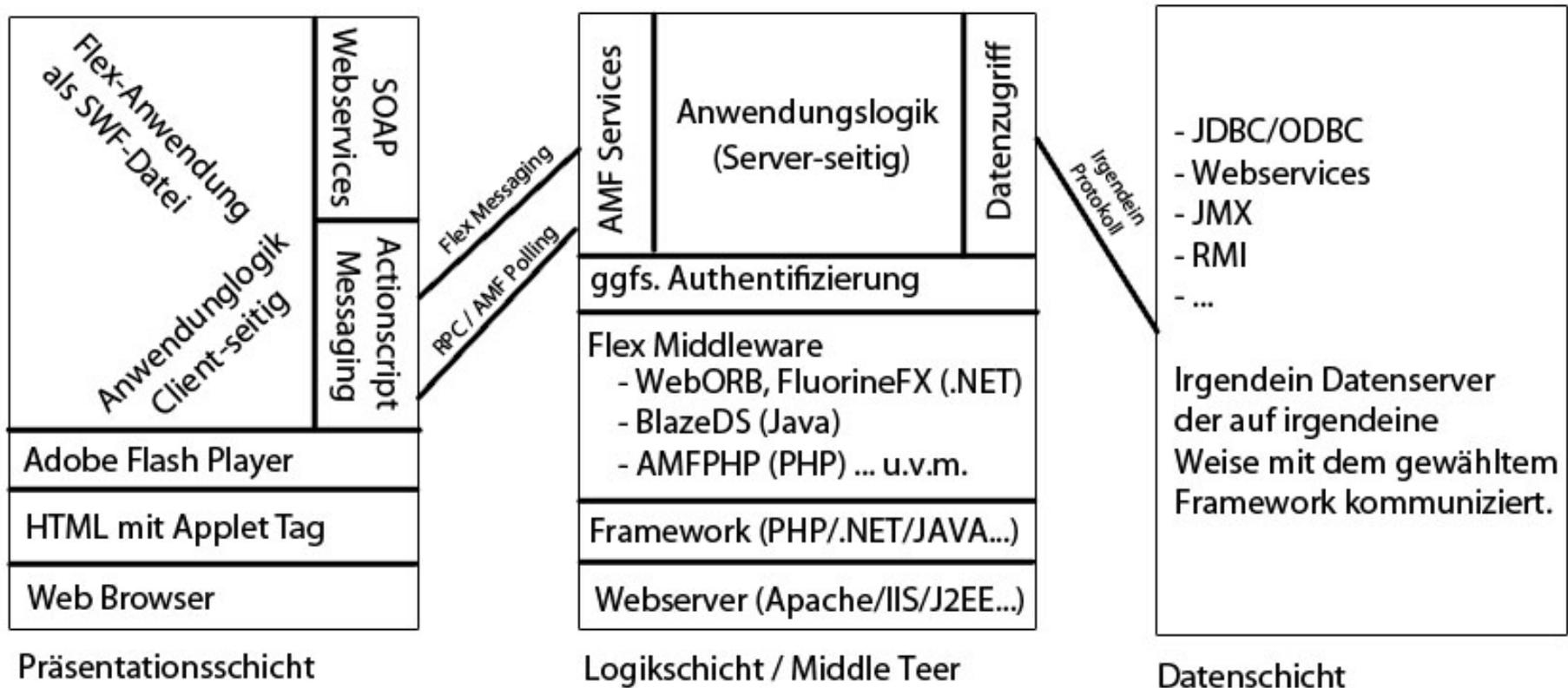
Was ist eine Rich Internet Application?



Was ist Adobe Flex?

- MXML-/Actionscript- und CSS-Compiler auf der Basis vom Flash Framework für die Visualisierung von Daten und Grafiken
- Rapid Application Development für das Web mit WYSIWYG-Editor
- Flex SDK ist Open Source und frei beziehbar
- Adobe verdient Geld mit Flex-Builder ...
 - ... als Eclipse Plug-In
 - ... als eigenständige Eclipse Version
- Laufzeitumgebung ist der Adobe Flash Player
- Datenanbindung (an Datenbank oder andere Datenquellen erfolgt per AMF (Actionscript Message Format) oder Webservices

Üblicher Aufbau einer Flex-Anwendung mit 3 Schichten (Three Tier Architecture)



Vorteile von Flex 1/2

- **Browser-unabhängige Programmierung**
 - keine lästigen Javascript-Workarounds
 - der Flash-Player interpretiert den Byte-Code
 - Flash ist wirklich abwärtskompatibel (Java!!)
- **Verteilung der Logik auf Frontend und Backendseite**
 - Besonders effizienter UI Aufbau
 - Verlagerung aufwendiger Berechnungen auf die Frontendseite
 - eigener Prozesskontext (nicht nur Sessions)
- **Große Community (Foren etc...)**
- **Verbreitung auch bei großen Herstellern betriebswirtschaftlicher Anwendungen (SAP mit Netweaver/Flex-Literatur)**

Vorteile von Flex 2/2

- Download der Anwendung bei Beginn, hierdurch einmalig längere Downloadzeit, anschließend schnelle binäre Kommunikation. (Browser-Cache beim zweiten Mal etc..)
- Offline-Verfügbarkeit mit Adobe AIR (z.B. Artikelstamm)
- Auch ohne Browser lauffähig
 - Zugriff aufs Dateisystem
 - Zugriff auf Peripherie (Bondrucker, Kassenschublade..)
- Flex ist nicht Flash
 - Es lässt sich wie ein herkömmliches Anwendungsframework programmieren
 - Es bietet einfache APIs für den Aufbau von Fenstern
 - Es bietet einfache Zugriffsmöglichkeiten auf Fremdsysteme
- Direktes Zeichnen von primitiven Shapes auf die "Leinwand" möglich (drawCircle, drawLine, etc ...)

Nachteile von Flex

- Mind. Flash-Player 9.0.0 für die jeweilige Plattform benötigt, bei Flash-Player 8 ohne Abfrage der Plugin-Installation bleibt der Bildschirm weiß
- Höhere Downloadzeiten zu Beginn
- Ohne Framework (wie Cairngorm) neigt man dazu die Logik unstrukturiert zwischen Client und Server zu verteilen.
- Actionscript lässt sich nicht typischer programmieren

Was ist
Adobe Flex?

Was kann man mit Adobe Flex machen?

Einsatzgebiete

Beispielanwendung

Wie erstellt man eine
Flex
Anwendung?

Signifikante
Merkmale

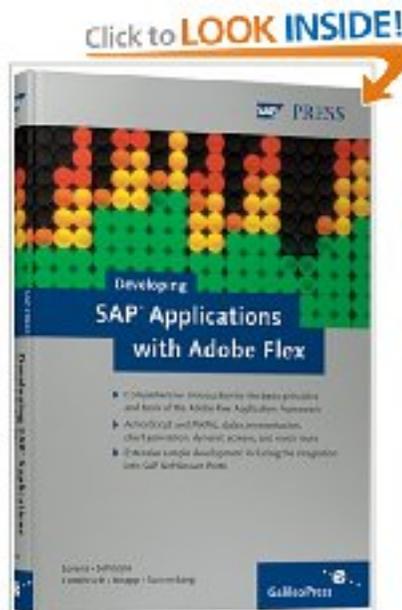
Einsatzgebiete

- Generell (nicht Hardwarenahe) Anwendungsentwicklung
 - Datenbankanwendungen
 - Abbildung von Workflows
 - Kollaborationsanwendung (Messaging!!!)
 - Abspielen von Audio Streams
 - Abspielen von Video Sequenzen
 - Abspielen von Flash Animationen

SAP Telefonat: "Wir setzen Flex immer zu

Marketing-
Haus-eigene
Dynpro) zu einfach

Zwecken ein, wenn unsere
Technologie (Web
wirkt."



Was ist Flex nicht?

- Flex ist nicht Flex (obwohl gleiches Framework)
 - Keine Grafik GUI für Zeichnung vorhanden
 - Bunte Animationen besser mit Flash
 - Tweenings (Motion Tween etc..) lassen sich nicht instanzieren wie in Flash (abgespeckt vom Framework)
 - Es wird keine Timeline verwendet, in die Actionscript "gequirlt" wird, sondern es kann direkt losprogrammiert werden.
- Flex ist kein Browser
 - Darstellung von HTML nur über Umwege
 - Z.B. DIV wird mit Javascript-API über die Flash-Anwendung gezeichnet
 - Diverse Komponenten können in Flex direkt HTML interpretieren, allerdings nie im Funktionsumfang eines Browsers

Beispielanwendung 1

- MySQL-Server
- PHP Backend
- AMFPHP als Middleware
- PHP Klassen als Services
- Authentifizierung

[Bsp 1: Loginmaske](#)

[Bsp 2: Datendarstellung mit Authentifizierung](#)

[Bsp 3: MDI Anwendung](#)

[Bsp 4: AMF PHP Browser](#)

Was ist
Adobe Flex

Was kann man mit
Flex machen?

Wie erstellt man eine Flex Anwendung?

Flex Builder

Backend Middleware

Graphic
Design

Komponenten zur Erstellung einer Flex Anwendung

- Entwicklungsumgebung für Flex
 - Flex Builder Plug-In für Eclipse
 - Flex Builder auf der Basis von Eclipse
 - Irgendein anderer Editor (ohne WYSIWYG-Funktion)
- Entwicklungsumgebung für Backend-Sprache
 - Java (Netbeans, Eclipse, JBuilder etc pp)
 - .NET (Visual Studio oder Mono DEV)
 - PHP (Eclipse mit PDT, Zend Studio, etc pp ...)
 - Cold Fusion
 - Ruby
 - u.v.m.
- Middleware für Action Script Messaging und Flex Remoting
 - .NET: FluorineFX, WebORB for .NET ...
 - Java: BlazeDS, WebORB for JAVA
 - PHP: AMFPHP, WebORB for PHP ...
 - Ruby: WebORB for Ruby ... u.v.m

Backend-Klasse in PHP u. AMFPHP 1/2

```
<?
include("BjoernsPHPSecurityClass.php");

class UserService extends BjoernsPHPSecurityClass {

    function UserService() {
        $this->methodTable = array(
            "getAllRegions" => array(
                "description" => "Gibt alle User zurueck",
                "access" => "remote",
                "returntype" => "recordSet",
                "roles" => "admin"
            )
        );
    }

    var $getAllUsersRoles = "admin,superuser";
    function getAllUsers() {
        return mysql_query("SELECT * FROM benutzer");
    }

}
?>
```

Backend-Klasse in PHP u. AMFPHP 2/2

```
<?php
class BjoernsPHPSecurityClass
{
    var $dbhost = 'localhost'; // your information here
    var $dbname = 'x';
    var $dbuser = 'x';
    var $dbpass = 'x';

    function BjoernsPHPSecurityClass()
    {
        // Initialize db connection
        $this->conn = mysql_pconnect($this->dbhost, $this->dbuser, $this->dbpass);
        mysql_select_db ($this->dbname);
    }

    public function beforeFilter($function_called)
    {
        $memberName = $function_called."Roles";
        return (@$this->$memberName) ? Authenticate::isUserInRole($this->$memberName) : true;
    }

    function login($email, $pass)
    {
        if (Authenticate::isAuthenticated()) return true;

        $result2 = mysql_query("SELECT * FROM benutzer WHERE email='".$email.'" AND password='".$pass.'");

        $j=0;

        while ($row = mysql_fetch_array($result2, MYSQL_NUM)) {
            $j++;
        }

        $isValidUser=($j>0);//test if valid user

        if ($isValidUser){
            Authenticate::login($user,"admin"); //todo:handle roles
            return true;
        }
        return false;
    }
}
?>
```

Backend-Klasse mit .NET und C# und FluorineFX Fluorine Templates und einfache .NET Klassen als Services

The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Code Editor:** Shows the `InboxService.cs` file with the following code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using System.Collections;
5 using System.Data.OleDb;
6 using System.Data.SqlClient;
7 using FluorineFx;
8
9 namespace OCTServices
10 {
11     [RemotingService("InboxService")]
12     public class InboxService
13     {
14         public ArrayList getLOUserInbox()
15         {
16             ArrayList contractinfo = new ArrayList();
17
18             SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=OCT;Integrated Security=True");
19             try
20             {
21                 conn.Open();
22                 SqlCommand comm = new SqlCommand("SELECT * FROM ContractInfo", conn);
23                 comm.Connection = conn;
24                 comm.CommandText = "SELECT * FROM ContractInfo";
25                 SqlDataReader reader = comm.ExecuteReader();
26                 while (reader.Read())
27                 {
28                     ContractInfoVO vo = new ContractInfoVO();
29
30                     vo.id = reader.GetValue(0);
31                     vo.creator = reader.GetValue(1);
```
- Project Explorer:** Shows the project structure for `OCTServices`, including files like `AuthenticationService.cs`, `CountryService.cs`, `CustomerService.cs`, `InboxService.cs`, `MailService.cs`, `MyConfigurationManager.cs`, and `OrderService.cs`.
- FluorineFX Templates Dialog:** A dialog box titled "Neues Projekt" is open, showing the "FluorineFx ServiceLibrary" template selected under "Von Visual Studio installierte Vorlagen". The dialog also shows the project name "ServiceLibrary1" and the location "H:\My Documents\Visual Studio 2005\Projects".
- Annotations:** Red boxes and arrows highlight the `[RemotingService("InboxService")]` attribute and the `FluorineFx ServiceLibrary` template, with labels "Erstellung von Services" and "FluorineFX Templates".

Backend-Klasse mit .NET und C# und FluorineFX

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Collections;
using System.Data.SqlClient;
using FluorineFx;
using System.Security.Principal;

namespace OCTServices
{
    [RemotingService("AuthenticationService")]
    public class AuthenticationService
    {
        // Ermittelt den Windows Benutzer über integrierte Windows-Authentifizierung im IIS Server (Single Sign
        On Intranet)
        public string getUser()
        {
            try
            {
                System.Security.Principal.WindowsPrincipal p =
                    System.Threading.Thread.CurrentPrincipal as System.Security.Principal.WindowsPrincipal;
                string strName = p.Identity.Name;
                string userWithDeletedDomainPrefix = strName.Substring(strName.IndexOf("\\") + 1).
                Trim();

                return userWithDeletedDomainPrefix;
            }
            catch (Exception e)
            {
                return e.Message;
            }
        }
    }
}
```

Backend-Klasse mit Java und BlazeDS

1.) Installation der BlazeDS.lib in die Anwendung oder fertigen Turnkey

Webserver (für Anfänger!!) nehmen

2.) Klasse schreiben:

```
public class MyRPC
{
    public String sayHello()
    {
        return "Hello from Remot Object!";
    }

    public int add(int a, int b)
    {
        return a+b;
    }
}
```

3.) Konfigurieren in *WEB-INF/flex/remoting-config.xml*

```
<destination id="myJavaClass" >
    <properties>
        <source>MyRPC</source>
        <scope>application</scope>
    </properties>
</destination>
```

Gutes Einsteiger-Tutorial: http://www.adobe.com/devnet/lifecycle/articles/blazeds_gettingstarted_04.html

Zugriff auf Backend-Services mit MXML <mx:RemoteObject>

```
<mx:Script>
  <![CDATA[
    private function saveOrderHandler(event:ResultEvent):void
    {
      var newOrderId:String = event.result.toString();
      orderService.saveOrderPositions(newOrderId, orderPositions);
    }

    private function faultHandler( event:FaultEvent ):void
    {
      Alert.show( event.fault.faultString, "Error" );
    }
    ...
  !>
</mx:Script>

<mx:RemoteObject id="orderService"
  destination="GenericDestination"
  OCTServices.OrderService"
  showBusyCursor="true"
  fault="faultHandler(event)" >
  <mx:method name="saveOrder" result="saveOrderHandler(event)" />
</mx:RemoteObject>
```

Zugriff auf Backend-Services mit Actionscript und RemoteObject

```
GetShipToCustomersByCountryDAO.as | OrderCreationTool.mxml | *Order.mxml | UserSettings.mxml | AddOrEditUserModalF
1 package controller.command
2 {
3     import mx.controls.Alert;
4     import mx.rpc.events.FaultEvent;
5     import mx.rpc.events.ResultEvent;
6     import mx.rpc.remoting.RemoteObject;
7
8     public class SendMailCommand
9     {
10         private var service:RemoteObject = new RemoteObject();
11
12         public function SendMailCommand(empfaenger:String, subject:String, message:String)
13         {
14             service.destination = "GenericDestination";
15             service.source = "OCTServices.MailService";
16             service.sendMail.addEventListener("result", onResult);
17             service.sendMail.addEventListener("fault", onFault);
18             service.sendMail(empfaenger, subject, message);
19         }
20
21         public function onResult(event:ResultEvent):void
22         {
23             var result:String = event.result.toString();
24             if(result=="ok")
25             {
26                 // OK
27             }
28             else
29             {
30                 // Meckern
31             }
32         }
33
34         public function onFault(event:FaultEvent):void
35         {
36             Alert.show(event.fault.toString(), 'ERROR!');
37         }
38     }
39 }
```

Was ist
Adobe Flex?

Was kann man mit
Flex machen?

Wie entwickelt
man Flex?

Signifikante Merkmale

Programmiersprache

Datenbindung
(Observer)

Layoutdesign

Die Programmiersprache

- Typsicher und nicht typsicher programmierbar
- Vollständig objektorientierte sowie prozedurale Entwicklung möglich
- Ohne Framework werden Projekte schnell unstrukturiert
 - Die Verwendung des Cairngorm-Frameworks, welches von Adobe mit den Kunden als Best Practices verkauft wird, bietet strukturierte Patterns
- Es gibt keine blockierenden Aufrufe - weder bei Dialogen (modale Dialog) noch bei sonstigen Abläufen. Jede Funktion wird als Thread gestartet
 - Reaktion auf das Ergebnis eines Aufrufs immer über Events
- Alle Remote Aufrufe (egal ob Webservices oder Flex-Remoting anfragen) sind asynchron

Datenbindung

- Komponenten werden immer an eine Datenquelle gebunden, hierbei handelt es sich i.d.R. um Arrays/ArrayCollections, die mit dem Tag [Bindable] vordefiniert sind
- Die ArrayCollections/Arrays werden über die dataProvider-Property der Komponente zugewiesen
- Ändert sich ein [Bindable]-getagtes Array, werden alle UI Komponenten aufgefordert sich zu aktualisieren
 - [Bindable] erfüllt die Funktionen eines Observer-Patterns
- Nach einem asynchronem Aufruf wird i.d.R. ein Array befüllt, woraufhin das Array diese Aktion allen UI Komponenten mitteilt

Layoutgestaltung

Flex-Anwendungen bestehen aus Vektorgrafiken und Pixelgrafiken. Die Grundkomponenten können umdesignt werden. Hierfür gibt es 2 Mechanismen

- Skinning (Starker Eingriff mit Änderung der Grafiken)
 - Scale-9-Format
 - Adobe Illustrator Templates
- Styles (Schwacher Eingriff nur Formatierungen)
 - CSS für die Farbgebung inklusive Vererbung
 - Zuweisung von Properties an den Tags

Danke noch Fragen?!

Björn Karpenstein

Handy: 0177 3313261

Email: bjoern.karpenstein@capri-soft.de