

.NET Framework, ASP.NET, C#

Oliver Krahulec

31.10.2006

SOFTEC

Obsah

- Úvod do .NET Frameworku
 - Architektúra, Model vykonávania
 - MS Visual Studio.NET
- ASP.NET
- ADO.NET
- DataSet objekty
- Webové služby
- Deployment – nasadzovanie aplikácií
- Je .NET správne riešenie?

2

SOFTEC

Úvod do .NET frameworku.

.NET Framework predstavuje aplikačnú platformu Microsoft novej generácie

- Umožňuje vytvárať:
 - Windows aplikácie
 - Windows služby
 - Webové aplikácie
 - Webové služby
 - Konzolové aplikácie
 - Iné

3

SOFTEC

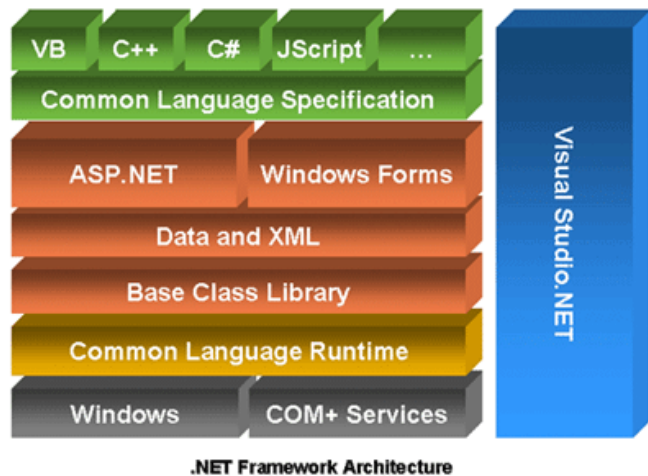
Kľúčové komponenty

- .NET Framework (<http://msdn.microsoft.com/net/>) sa skladá z dvoch komponentov:
 - CLR (Common Language Runtime), ktorého JITCompiler kompiluje a vykonáva .NET kód
 - FCL (Framework Class Library), knižnica v rádo stovkách tried na ktorej .NET Framework závisí.

4

SOFTEC

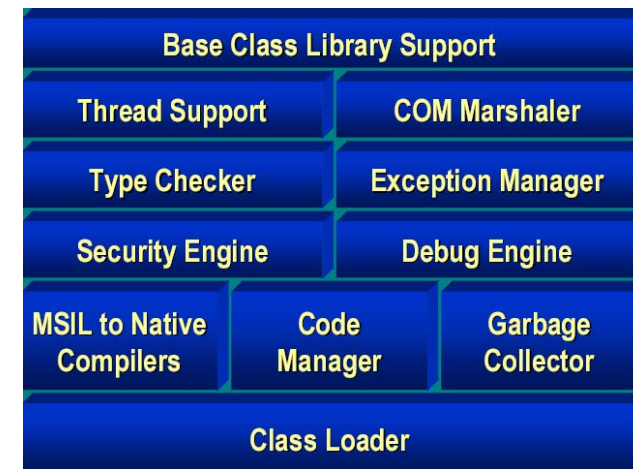
.NET Framework architektúra



5

SOFTEC

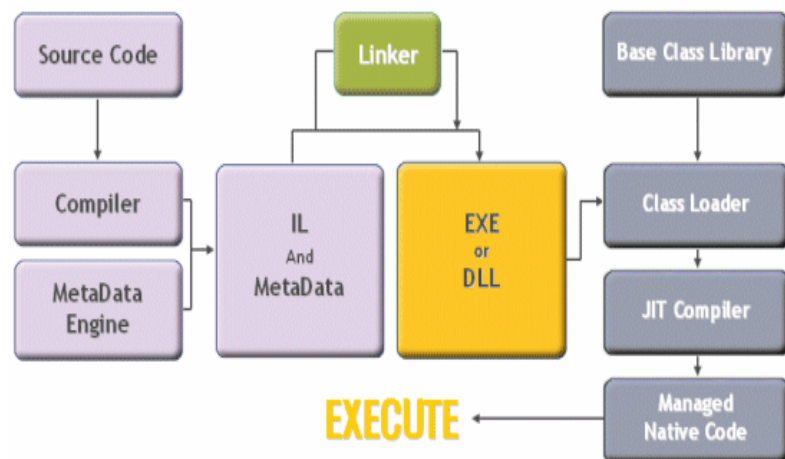
Common Language Runtime - CLR



6

SOFTEC

Execution model of .NET



7

SOFTEC

Microsoft .NET Visual Studio

- Visual Studio .NET poskytuje nasledovné vlastnosti mimo SDK:
 - Plne integrované, vizuálne vývojové prostredie
 - Popri štandardnej funkčnosti poskytuje možnosti pre vytvorenie:
 - Setup a deployment projektov
 - Help súbory
 - Remote debugging

8

SOFTEC

Základný .NET model

- Plne objektovo orientovaný, prevzatý prevažne z technológie Java
 - Triedy sú organizované do funkčných grúp nazývaných namespaces
 - Jednoduchá dedičnosť + podpora pre rozhrania
- Neexistuje "DLL Hell"
 - DLL ("assemblies") sú umiestnené lokálne do vašej aplikácie (v adresári bin); alebo
 - Sú nainštalované v Global Assembly Cache
 - Viaceré verzie DLL môžu existovať v cache
- Bohatá množina komponentov
- Základné .NET služby sú zozbierané v System namespace (System.IO, System.NET atď.)
- Programátor môže vytvoriť vlastné namespaces alebo derivovať triedy z iných
- Multiplatformové aplikácie

9

SOFTEC

Základný .NET model

- Integrovaná, robustná podpora inštalovania aplikácií (bez nutnosti Wise alebo InstallShield)
- Nie je potrebná práca s registrami
- Typovo bezpečné programovanie
- Zlepšenie aplikačnej bezpečnosti
- Jednotný model spracovania výnimiek
- Jazyková nezávislosť
- Spoločné vývojové prostredie umožňuje ladenie viacerých jazykov
- Rozšírené možnosti pre znovu použitia kódu

10

SOFTEC

.NET z pohľadu bezpečnosti

- Bezpečnosť je základnou časťou .NET.
- Znamená zabezpečiť:
 - Bezpečný typový systém
 - Kontrola autenticity kódu - signature
 - Autorizácia prístupu ku zdrojom
 - Deklaratívna bezpečnosť
- Ostatné bezpečnostné modely:
 - Prístupové politiky
 - Role
- Kryptografické metódy sú prístupné aplikáciám.

11

SOFTEC

ASP.NET pre webové aplikácie

- ASP.NET (.aspx, .asmx, or .ascx)
 - Stránky pozostávajú z HTML spolu s množstvom aspx "controls" prvkov pre generovanie formuláru, elementy stránky a spoločné validácie
 - Manipulácia s týmito prvkami je možná pomocou objektovo orientovaného modelu, alebo pomocou nastavenia vlastností v HTML kóde
 - Kód môže byť vytváraný pomocou jazykov C#, VB.NET, J#, C++
 - Prístupné sú stovky tried, ktoré sú rovnaké aj pri vývoji Windows aplikácií
 - Vývoj je veľmi podobný ako pri Windows aplikáciám
 - Význačná podpora pre tvorbu webových služieb

12

SOFTEC

ASP.NET pre webové aplikácie

- ASP.NET Typy súborov
 - .aspx: webová aplikácia (známa tiež ako webový formulár - stránka)
 - .asmx: webová služba - .NET 1.1
 - .ascx: User controls (znovu použiteľný blok kódu ASP.NET, ktorého funkčnosť sa podobá „server-side include“ príkazom)

13

SOFTEC

ASP.NET komponenty

- HTML prvky
 - Odzrkadľuje štandardné HTML form elements
 - Prvky sú programovo prístupné v „codebehind“
 - Neukladajú svoj stav
 - Nepodporujú priame napojenie na dátový zdroj

14

SOFTEC

ASP.NET komponenty

- Prvky Web servera
 - Robustnejšie, predstavujú webové formulárové prvky
 - Prvky sú spracovávané a vykonávané na aplikačnom serveri
 - Programovo prístupné z „codebehind“ tried
 - Existuje možnosť ukladania ich stavu pomocou objektu viewstate /.NET 2.0 priniesol controlstate
 - Podporujú priame napojenie na dátový zdroj
 - Existuje možnosť vyvolania postback udalosti, kontrolné a ostatné udalosti

15

SOFTEC

ASP.NET komponenty

- Validačné prvky - controls
 - Jednoduché prvky pre kontrolu užívateľských vstupov
 - RequiredFieldValidator
 - CompareValidator
 - RangeValidator
 - RegularExpressionValidator
 - CustomValidator
 - ValidationSummary

16

SOFTEC

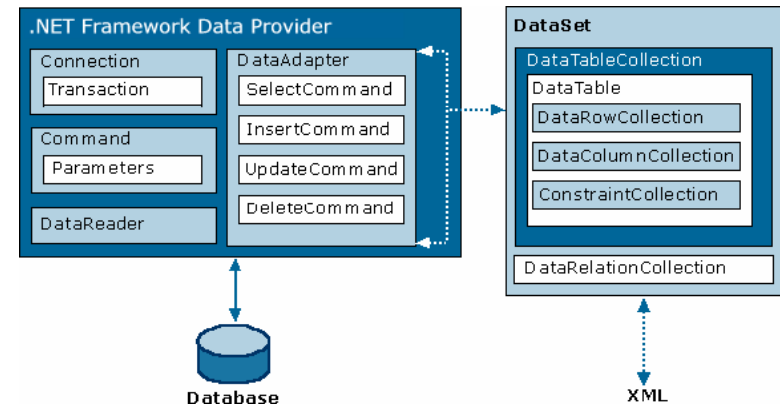
ADO.NET

- .NET Framework Data Provider poskytuje triedy, ktoré implementujú potrebné rozhrania. Zabudovaný poskytovateľ v Framework 1.1 zahŕňajú:
 - SqlClient (pre SQL Server 7.0 a novšie)
 - OracleClient
 - OleDb
 - Odbc
- Pripojovacie triedy pre napojenie na databázu
- Dáta sa získajú pomocou DataReader alebo DataAdapter objektov
 - DataReader poskytuje sekvenčný prístup, prevažne používaný na priame naplnenie premenných alebo prvkov
 - DataAdapter naplní objekt DataSet s dátami a synchronizuje všetky zmeny v objekte s databázou

17

SOFTEC

ADO.NET architektúra



18

SOFTEC

ADO.NET

- Použite radšej konkrétnu implementáciu adaptéra na DB ako ODBC
- DataSet, nie RecordSet
 - Odpojené od DB
 - Viaceré tabuľky, spájajú ho DataRelations
 - Možnosť čítania/zápisu XML a XML schémy
- Možnosti parametrizovania SQL dotazov
- Čas pripojenia je minimalizovaný

19

SOFTEC

DataSet objekty

- DataSet:
 - Multi tabuľka, dáta sú odpojené od dátového zdroja
 - Existujú v odpojenom stave – bez nepretržitého spojenia do databázy
 - Sú napĺňané data adaptérom
 - Zmeny v datasete sú synchronizované s databázou pomocou data adaptéru
 - Môžu automaticky generovať XML súbory a XML schémy a taktiež ich čítať
 - Komponenty (prezentačnej vrstvy) typicky používajú ako dátový zdroj DataSet alebo DataView z DataSetu

20

SOFTEC

Webové služby – Web services

- Aplikácie, ktorých metódy sú vystavené pre vzdialené volanie (spúšťanie)
- Tieto metódy sa typicky spúšťajú pomocou SOAP (Simple Object Access Protocol) cez HTTP
- WSDL (Web Services Description Language) popisujú webové služby
- UDDI (Universal Description, Discovery, and Integration) pomáhajú lokalizovať a integrovať aplikácie spolu s webovými službami

21



Webová služba

```
/// <summary>
/// Summary description for WebService
/// </summary>
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class WebService : System.Web.Services.WebService
{

    public WebService()
    {

        //Uncomment the following line if using designed components
        //InitializeComponent();

    }

    [WebMethod]
    public string HelloWorld()
    {

        return "Hello World";

    }

}
```

22



Deployment - inštalácia

- 4 typy inštalčných projektov (iba ich pridáme k riešeniu):
 - Setup Project (pre inštalovanie Windows apl.)
 - Web Setup Project (pre inštalovanie Web apl. na Windows serveri)
 - Merge Module (pre vytvorenie balíka komponentov, ktoré budú zdieľané viacerými aplikáciami)
 - Cab Project (používaný na inštaláciu ActiveX prvkov)
- Rôzne pohľady:
 - File System Editor
 - Registry Editor
 - File Types Editor
 - User Interface Editor
 - Custom Actions Editor
 - Launch Conditions Editor
- Vytváranie inštalčných balíkov je vytvorené pomocou Release módu

23



Je .NET správne riešenie?

- Pre vývojárov, takmer určite ak:
 - Potrebujete vytvoriť alebo používať webové služby
 - Vaša aplikácia vo veľkej miere pracuje s XML
 - Chcete pracovať s dátovým modelom prístupu typu "disconnected dataset"
 - Tvoríte web aplikácie a požadujete viac objektovo orientovaný model vývoja aplikácie
 - Vaši klienti si to želajú ☺

24



Je .NET správne riešenie?

- Vo všeobecnosti pre obchod, takmer určite ak:
 - Je potrebná schopnosť integrovať aplikácie dodávateľov, partnerov alebo zákazníkov.
 - Stráca sa príliš veľa prostriedkov pri „znovu vynájdení kolesa“ zakaždým, keď je vytváraná webová aplikácia a požadujú sa jednoduchšie a znovu použiteľné prezentačné a aplikačné elementy

25

SOFTEC

Je .NET správne riešenie?

- Možno nie pre vývojárov ak:
 - Vaši klienti to od Vás nežiadajú
 - Ste zaplavený s prácou pre vývoj vo VB, C++, alebo akýkoľvek iný jazyk v ktorom vyvíjate
- Možno nie pre obchod ak:
 - Nepotrebuje integrovať vaše aplikácie s dodávateľmi, partnermi alebo klientmi
 - Vaše webové aplikácie sú relatívne jednoduché a máte iba nízke nároky na znovu - použiteľnosť

26

SOFTEC

C# 2.0 novinky

- Generics
- Iterators
 - Zjednodušenie použitia ako príkaz iteruje medzi prvkami kolekcie
- Nullable Types
- **Partial Class**
 - Čiastočná definícia typu umožňuje rozdeliť deklaráciu jedného typu, napríklad triedy do viacerých súborov. Visual Studio Designer používa túto vlastnosť na oddelenie generovaného od používateľského kódu.
- **Anonymné metódy**
- **Statické triedy**

27

SOFTEC

Generics

- rozšírenie o koncept typových parametrov, ktoré umožňujú odložiť deklaráciu jedného alebo viacerých typov triedy alebo metódy, až pokiaľ nie sú deklarované alebo vytvárané aplikačným kódom. Typickým príkladom môže byť vytvorenie typovej kolekcie, poľa.

```
class FakturaCollection : Collection<Faktura>
{
}
}
```

28

SOFTEC

Generics - príklad

```
// Declare the generic class
public class GenericList<T> {
    void Add(T input) { }
}

class TestGenericList {
    private class ExampleClass { }
    static void Main() {
        // Declare a list of type int
        GenericList<int> list1 = new GenericList<int>();
        // Declare a list of type string
        GenericList<string> list2 = new GenericList<string>();
        // Declare a list of type ExampleClass
        GenericList<ExampleClass> list3 = new GenericList<ExampleClass>();
    }
}
```

79

Generics - výhody

- Maximalizovanie znovu použiteľnosti kódu, typová bezpečnosť, zlepšenie výkonnosti (boxing, un-boxing operácie)
- .NET Framework class library prináša niekoľko generic kolekcií. Nachádzajú sa v [System.Collections.Generic](#) namespace.
- Môžete si vytvoriť vlastné generic rozhrania, triedy, metódy delegátov, udalosti.
- Informácie o typoch použitých pomocou generic data type je možné získať v run-time režime pomocou prostriedkov reflexie

Iterators

- Rozhranie IEnumerator

```
public interface IEnumerator
{
    object Current { get; }
    bool MoveNext();
    void Reset();
}
```

Iterators

```
public class DaysOfTheWeek : System.Collections.IEnumerable {
    string[] m_Days = { "Sun", "Mon", "Tue", "Wed", "Thr", "Fri", "Sat" };

    public System.Collections.IEnumerator GetEnumerator() {
        for (int i = 0; i < m_Days.Length; i++) {
            yield return m_Days[i];
        }
    }
}

class TestDaysOfTheWeek {
    static void Main() {
        // Create an instance of the collection class
        DaysOfTheWeek week = new DaysOfTheWeek();

        // Iterate with foreach
        foreach (string day in week) {
            System.Console.Write(day + " ");
        }
    }
}
```


Nullable typy

- Null typy sú inštanciami štruktúry [System.Nullable](#).
- Napríklad [Nullable<Int32>](#), znamená "Nullable of Int32," a môže reprezentovať hodnotu v rozsahu od -2147483648 do 2147483647, alebo hodnota môže byť **null**.
- Syntax **T?** je skrátenie [System.Nullable<T>](#), kde **T** je dátový typ.
- Property [System.Nullable.GetValueOrDefault](#) vráti hodnotu, alebo default hodnotu daného typu, ak hodnota je **null**, napríklad

```
int j = x.GetValueOrDefault();
```
- Read-Only vlastnosti [HasValue](#) a [Value](#) sa používajú pre zisťovanie hodnoty alebo null hodnoty, napríklad

```
if(x.HasValue) j = x.Value;
```
- Vlastnosť **Value** vráti hodnotu alebo generuje výnimku [System.InvalidOperationException](#) ak hodnota nie je inicializovaná.
- Operátor **??** sa používa na priradenie default hodnoty, keď nullable dátový typ je priradený do normálneho dátového typu a jeho hodnota je, napríklad

```
int? x = null; int y = x ?? -1;
```

33

SOFTEC

.NET 3.0 – Prehľad

- *ASP.NET 3.0 = ASP.NET 2.0 + nové nástroje a technológie*
- [XAML – Extensible Application Markable Language](#) – riadenie, konfigurovanie a definovanie aplikačných komponentov a controls-ov pomocou xml.
- [Windows Communication Foundation](#) – množina API pre protokoly/integráciu/konektivitu k zariadeniam, enterprise systémom, priemyselným technológiám
- [Windows Workflow Foundation](#) – modelovanie aktivít, workflow a spúšťanie workflow
- [Card Space](#) – certifikácia a autorizácia – len od Windows Vista
- [IIS 7.0](#) – komponentuálne zloženie a zostavenie IIS podľa potrieb – len od Windows Vista
- ASP.NET „[ATLAS](#)“ – framework integrujúci AJAX technológiu do .NET 2.0
- [Office OpenXML](#) – Office2007 dokumenty sú založené na XML formáte a prístupné cez OpenXML API
- [Sharepoint 3.0](#) – plne kompatibilný a do .NET 2.0 reimplementovaný nástroj na vytváranie aplikácií a site-s portálového typu
- Produkt bude oficiálne vydaný koncom roka 2006

34

SOFTEC

WCF – Windows communication Foundation

- Distribuované transakcie dnes:
 - [Rôzne technológie](#)
 - WSE, Remoting, RMI, EJB, COM+, JMS...
 - [Navzájom nekompatibilné](#)
 - Vývoj je závislý na zvolenej technológii
 - Pri zmene technológie – veľký zásah do app
- WCF ktoré rieši distribuované spracovanie.
- Je založené na základe kategórií WS.

35

SOFTEC

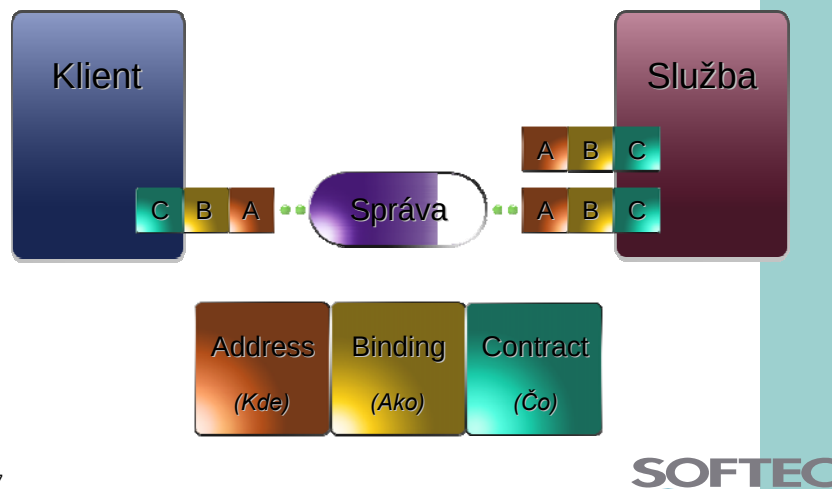
WCF – základný koncept

- **Address**
 - [Adresa, kde služba beží](#)
- **Binding**
 - [Zpôsob komunikácie služby](#)
 - [Ako sa napojiť \(smery komunikácie\)](#)
- **Contract**
 - [Poskytované rozhranie \(metódy, dáta\)](#)
 - [Nezávislé na AB](#)
 - [Pravidlá a obsah správ](#)

36

SOFTEC

Address, Binding, Contract



37

SOFTEC

Contract - typy

- Service
 - Popisuje poskytované služby
 - Service Contract - Typy
 - Request - Response
 - OneWay
 - Bez odpovedi
 - Duplex
 - Asynchronní metody
 - Nelze použít pro všechny typy Bindings
 - Fault
 - Chyba, kterou nelze ošetřit na serveru
- Data
 - Popisuje strukturu poskytovaných dat
- Message
 - Modifikuje strukturu SOAP správy
 - Instrukcie vykonáva potom infraštruktúra/server

38

SOFTEC

Binding - prehľad

	Interop.	Security	Session	Transactions	Duplex
BasicHttpBinding	BP 1.1	T			
WsHttpBinding	WS	T S	X	X	
WsDualHttpBinding	WS	T S	X	X	X
NetTcpBinding	.NET	T S	X	X	X
NetNamedPipesBinding	.NET	T S	X	X	X
NetMsmqBinding	.NET	T S	X	X	
MsmqIntegrationBinding	.NET	T			
NetPeerTcpBinding	.NET	T S			X

T = Transport Security | S = WS-Security Message Security

39

SOFTEC

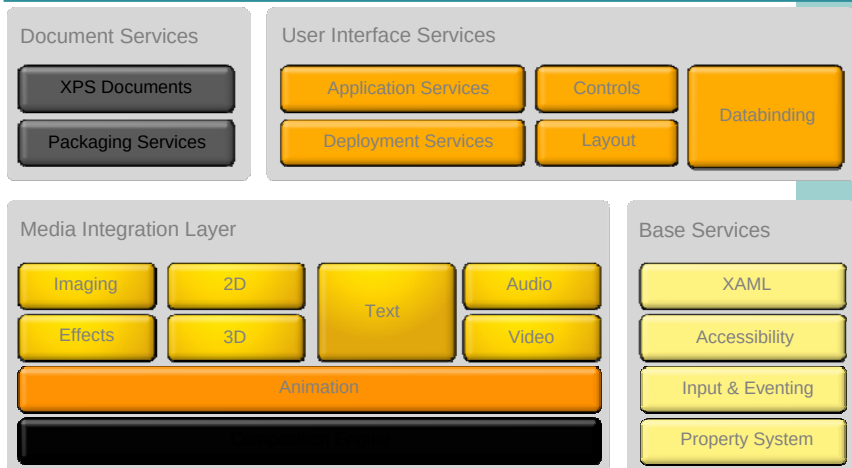
WPF – Windows Presentation Foundation

- Technológia pre vytváranie ťažkých GUI aplikácií pre Ms Windows Vista.
- Jednotný prístup k UI, dokumentom a médiám
- Z web prístupu zavádza do tvorby GUI deklaratívny aplikačný model prostredníctvom XAML. Preto prístup k tvorbe takýchto aplikácií je veľmi podobný prístupu pre tvorbu web aplikácií v ASP.NET 3.0.
- Prínosy:
 - Pružný model GUI
 - Dátová predikcia
 - Šablóny obsahujú vzhľad aj dynamiku
 - Grafika je prístupná deklaratívne(XAML) aj programovo
 - 2D aj 3D modely, vektorové, s animáciami

40

SOFTEC

WPF - komponenty



41

SOFTEC

Web 2.0 ATLAS/AJAX

- ATLAS premenovaný na Microsoft AJAX Library (cca mesiac)
 - Aktuálna verzia 1.0 beta 1 (v skutočnosti ťažká alfa ☹)
- ATLAS je MS framework na použitie AJAX technológie pre vytváranie komplexného UI vo web aplikáciách v ASP.NET. Základnou ideou je, že sa posiela/reloaduje/meni/komunikuje len časť web stránky.
- 1995 – prvé princípy AJAX – iframe so zdrojmi, ktoré sa čítajú postupne
- AJAX nie je technológia, je to vhodné spojenie dnes na webe bežne používaných technológií:
 - XHTML, CSS
 - DOM
 - XMLHttpRequest
 - XML
- ATLAS - AJAX pre .NET, t.j. prináša OO a .NET princípy, triedy, namespaces, enum, dedičnosť, interfaces
- ATLAS - dva typy prístupu
 - Javascript – tradičný
 - Deklaratívny model podobný ASP.NET

42

SOFTEC

Web 2.0 ATLAS/AJAX

- Jedným z dôvodov, ktoré Microsoft viedli k vytvoreniu takéhoto frameworku bol fakt, že vývoj AJAX-enabled web aplikácií je problematický najmä z hľadiska následnej funkčnosti aplikácie v rôznych prehliadačoch a na rôznych platformách. Tento cieľ sa zatiaľ (Jul 2006 CTP) Atlasu absolútne nedarí naplniť. Je plne kompatibilný jedine s Microsoft Internet Explorerom a čiastočne kompatibilný s Gecko browserami. Celý zvyšok spektra je nepodporovaný, z čoho sa dá usudzovať, že ostré nasadenie Atlasu v rámci projektov zatiaľ nepripadá do úvahy.
- Na druhej strane však treba oceniť snahu navrhnuť framework tak, aby sa vývoj Ajax-enabled web aplikácií príliš nelíšil od tradičného vývoja v ASP.NET.

43

SOFTEC