

C# Programmierung

Eine Einführung in das .NET Framework

Tag 7

•••

Prozesse und Möglichkeiten

Wofür C# (und das .NET-Framework) noch gut sind und welche Möglichkeiten sich ergeben

Die Process Klasse

- Kennen wir schon: *Starten eines Prozesses* erfolgte über diese Klasse!
- Können damit aber auch vorhandene Prozesse kontrollieren und manipulieren (siehe *TaskManager*)
- Können neue Prozesse starten und Ausgaben umleiten
- Zugriff auf Threads über deren *ProcessID*

Callbacks



- Rückruf über ein Ereignis
- Häufig verwendet bei asynchronen Methoden in .NET
- Ältere .NET Funktionen geben außerdem einen **Handler**, welcher Informationen zur async Methode besitzt
- Oftmals kann man wählen: **sync** oder **async**?

Beispiel

01 – Process Callbacks

Spieleentwicklung

- **Kaum** Performanceunterschied von managed Code (z.B. C#) zu unmanaged Code (z.B. C++)
- DirectX SDK bringt *managed DirectX fürs Visual Studio*
- OpenGL über *Wrapper* wie z.B. SharpGL möglich – oder eigene Implementierung denkbar
- **Demo** von DirectX mit C# in einer 3D Umgebung

Beispiel

02 – DirectX mit C#

Dynamische Webseiten

- Durch Kompilierung sehr viel schnellere und mächtigere Seiten möglich als mit Skriptsprachen
- Programmierung in Visual Basic oder C# möglich – gute **Trennung** von Code und Layout
- Umdenken im Bezug auf WinForms: Hier *instanzloser* Code der nur am (Windows) **Server ausgeführt** wird
- Kleines **Beispiel** einer ASP.NET Seite

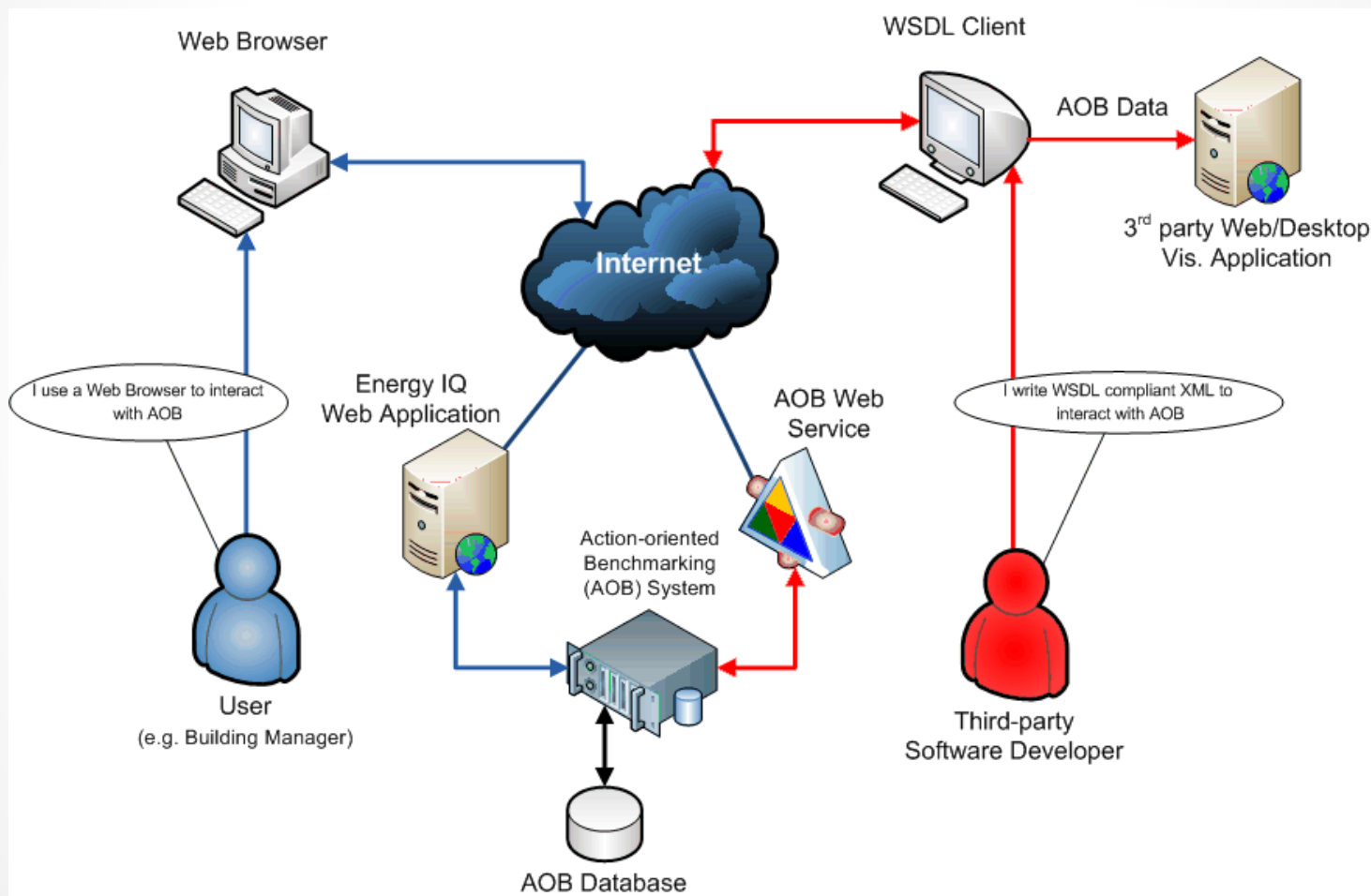
Beispiel

03 – Webseiten mit ASP

Webservices

- Funktionen die im Intra- / Internet verfügbar sein sollen, z.B. um Daten (von einem Server) abzurufen
- Direkte Unterstützung im .NET für diese wichtige Technologie
- Austausch der Daten im XML Format – Sicherheitskonzept für sensible Daten notwendig
- Eine kurze **Demonstration** eines Webservices

Aufbau eines Webservice



Beispiel

04a – Webservice [Client]

04b – Webservice [Server]

C# 4.0 mit VS 2010

- Neue Funktionen sollen v.a. Multiprozessor Systemen helfen
- Threads wurden **abgeschafft** und durch Tasks ersetzt (*im OS sind natürlich weiterhin Threads*)
- Sehr leichte Aufspaltung durch anonyme Funktionen (Lambda expressions) und Parallel Ext.
- Demo einer *Parallel.For()* Ausführung im VS 2010

Beispiel

05 – Parallel in C# 4

WindowsPhone 7

- Kostenloses SDK verfügbar
- VS 2010 notwendig oder VS 2010 Express Edition (kostenlos – gibt es auch mit WP7 SDK integriert)
- Möglichkeiten der Distribution von Anwendungen
- Kostenlose Mitgliedschaft als MS Entwickler – wie?
- Kurze **Demo App** für WP 7

Beispiel

06 - WeatherForecast