



Microsoft .NET

InfoPoint 8. Juni 2005

Stefan Bühler



Inhalt

- Was ist .NET
- Was steckt dahinter
- Warum ist .NET so wie es ist
- Die Säulen von .NET
- .NET Framework 2.0 / VisualStudio 2005
- Beispiel
- Referenzen & Links
- Fragen (und Antworten)



Was ist .NET (1)

- Technologie?
 - Nein, aber umfasst viele Technologien (ADO.NET, CLR, ...)
- Produkt?
 - Nein, aber umfasst zahlreiche Produkte (Visual Studio.NET, Exchange 2000 Server)
- Software-Architektur
 - Nein, macht kaum explizite Aussagen über Architekturen, impliziert aber durch verschiedene Technologien architektonische Aspekte



Was ist .NET (2)

■ Plattform?

- Sammlung sich ergänzender Technologien, für die man Software entwickeln kann → .NET ist Plattform

■ Strategie?

- Strategie: Summe von Massnahmen zur Erreichung eines Zieles → .NET ist Strategie



Was steckt dahinter (1): Analyse

- Entwicklung weg von Rich Clients hin zu Thin Clients
- zunehmende Miniaturisierung von Computern
- Rückwärtskompatibilität hindert Entwicklung
- Technologie-Ansätze wie Component Object Model COM haben sich zu schwer wartbaren Softwaregebilden entwickelt

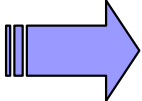
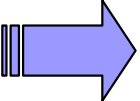
Was steckt dahinter (2): Analyse

- COM Objekte global auf Rechner verfügbar → ‚DLL-Hell‘
 - Windows-Bindung der Technologien, keine direkte Unterstützung von anderen Plattformen (CORBA / J2EE)
 - Datenaustausch nur über COM Typsystem
 - zu „schwergewichtige“ APIs
 - keine durchgängige OO der APIs
- durchgängig vernetzte, sehr heterogene Hardwarewelt bedroht Microsofts Geschäftsmodell (Verkauf von SW-Paketen für eine sehr kleine HW-Bandbreite)

Was steckt dahinter (3): Konsequenzen

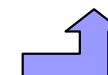
- Plattform für Nutzung und Angebot von Komponenten-DL im Internet
- Wie kann Microsoft in einer Welt von weniger SW-Paketen und mehr SW-DL bestehen
- Auf Standards setzen und Standards definieren helfen (Interoperabilität)
- Leichtgewichtige und ‚state-of-the-art‘ Programmierschnittstelle → zügige Portierung auf eine Vielzahl von Plattformen
- Lose Kopplung von Komponenten über das Internet wie auch möglichst einfache lokale Installation von Komponenten auf unterschiedlichsten HW-Plattformen

Warum ist .NET so wie es ist (1)

- Interoperabilität zwischen Plattformen
 - XML als Datenaustausch-Format
- Kommunikation zwischen Internet-DL
 - Webservices
 - Simple Object Access Protocol SOAP
- Webservices beschreiben und finden
 - Web Service Description Language WSDL 
 - Universal Description, Discovery and Integration UDDI 

WebService Description Language WSDL

```
<?xml version="1.0"?>
<definitions name="StockQuote"
  <types>
    <schema targetNamespace="http://example.com/stockquote.xsd" xmlns="http://www.w3.org/2000/10/XMLSchema">
      <element name="TradePriceRequest">
        <complexType>
          <all>
            <element name="tickerSymbol" type="string"/>
          </all>
        </complexType>
      </element>
      <element name="TradePrice">
        <complexType>
          <all>
            <element name="price" type="float"/>
          </all>
        </complexType>
      </element>
    </schema>
  </types>
  <message name="GetLastTradePriceInput">
    <part name="body" element="xsd1:TradePriceRequest"/>
  </message>
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput"/>
      <output message="tns:GetLastTradePriceOutput"/>
    </operation>
  </portType>
  <binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetLastTradePrice">
      <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>
```



Universal Description, Discovery and Integration UDDI

Microsoft UDDI Business Registry (UBR) node - Microsoft Internet Explorer

Adresse <http://uddi.microsoft.com/search/frames.aspx> Wechseln zu

Microsoft uddi Business Registry Node MS Products | MS Search | MSDN Home | Microsoft Home Sign In

Home Search Quick Help UDDI Help Go

Search

Results Explorer

- www.dotnet.be
 - Bart De Smet
 - Gunther Beersaerts
 - Fibonacci
 - <http://www.dotnet.be>
 - Flavius
 - <http://www.dotnet.be>
 - ISBN & EAN
 - <http://www.dotnet.be>

www.dotnet.be | ISBN & EAN
<http://www.dotnet.be/websvc/isbn.asmx>

A binding represents an access point and one or more instances of the service that can be accessed at that point. An Instance Info typically points to an interface description or specification.

Details Instance Infos

Details present the access point, protocol supported, and a brief description of the binding. The binding key is unique and is intended for use in programmatic queries.

Access Point:
<http://www.dotnet.be/websvc/isbn.asmx>

URL Type:
Http

Binding Key:
59b8b3f3-41a3-429b-a149-5134b39c99a6

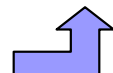
Descriptions

(en)
Webservice for ISBN & EAN code validation

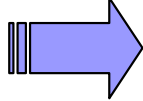
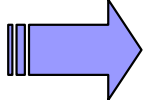
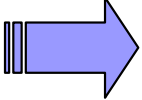
1 record(s) found.

© 2000-2003 Microsoft Corporation. All rights reserved.
[Microsoft Terms of Use](#) | [UDDI Terms of Use](#) | [Privacy Policy](#)

Internet



Warum ist .NET so wie es ist (2)

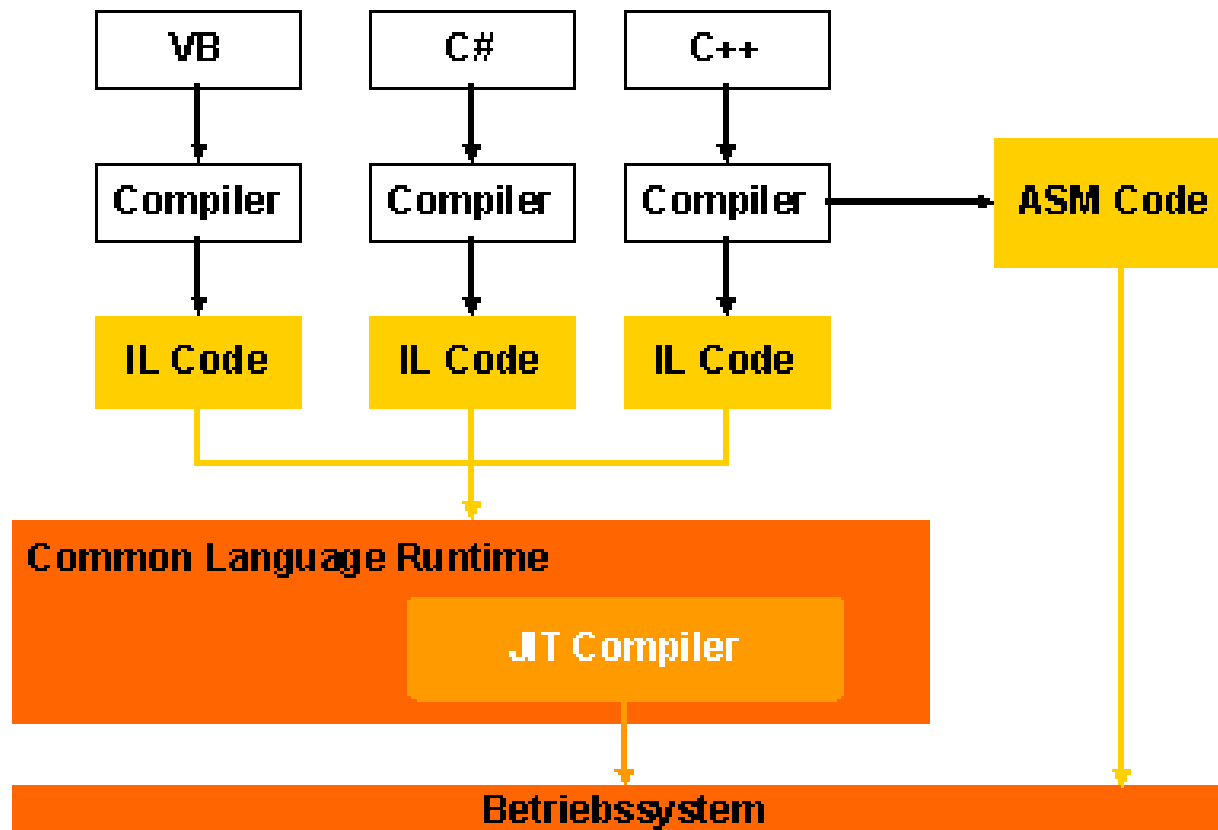
- einheitliche Schnittstelle für viele Plattformen
 - Common Language Runtime CLR 
- Plattformunabhängiger Code
 - Microsoft Intermediate Language MSIL 
- Interoperabilität zwischen Programmiersprachen
 - Common Type System CTS 
- Vereinfachte Installation von Software
 - Assemblies



Common Language Runtime CLR (1)

- 1 Runtime für n Programmiersprachen
- JIT Übersetzung von MSIL in Maschinencode
- Speicherverwaltung
- Security

Common Language Runtime CLR (2)



Common Language Runtime CLR (3)

- Perl
- Fortran
- Python
- Eiffel
- Oberon
- Cobol
- Haskell
- ...





Microsoft Intermediate Language

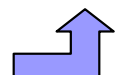
MSIL (1)

- Managed Code, enthält Meta-Informationen
- Wird JIT in prozessorspezifischen Maschinencode übersetzt (auf Methodebene), nie interpretiert

Microsoft Intermediate Language

MSIL (2)

```
DefaultTraceListener::StackTraceToString : class System.String(class [mscorlib]System.Diagnosto...
.method private hidebysig instance class System.String
  StackTraceToString(class [mscorlib]System.Diagnostics.StackTrace trace,
    int32 startFrameIndex,
    int32 endFrameIndex) il managed
{
  // Code size          302 (0x12e)
  .maxstack 3
  .locals (class [mscorlib]System.Text.StringBuilder V_0,
    int32 V_1,
    class [mscorlib]System.Diagnostics.StackFrame V_2,
    class [mscorlib]System.Reflection.MethodBase V_3,
    class [mscorlib]System.Reflection.ParameterInfo[] V_4,
    int32 V_5,
    class [mscorlib]System.Reflection.ParameterInfo V_6,
    int32 V_7)
  IL_0000: ldc.i4      0x200
  IL_0005: newobj      instance void [mscorlib]System.Text.StringBuilder::.ctor(int32)
  IL_000a: stloc.0
  IL_000b: ldarg.2
  IL_000c: stloc.1
  IL_000d: br         IL_0114
  IL_0012: ldarg.1
  IL_0013: ldloc.1
  IL_0014: callvirt   instance class [mscorlib]System.Diagnostics.StackFrame::get_MemberInfo()
  IL_0019: stloc.2
  IL_001a: ldloc.2
  IL_001b: callvirt   instance class [mscorlib]System.Reflection.MethodBase::get_DeclaringType()
}
```

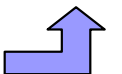
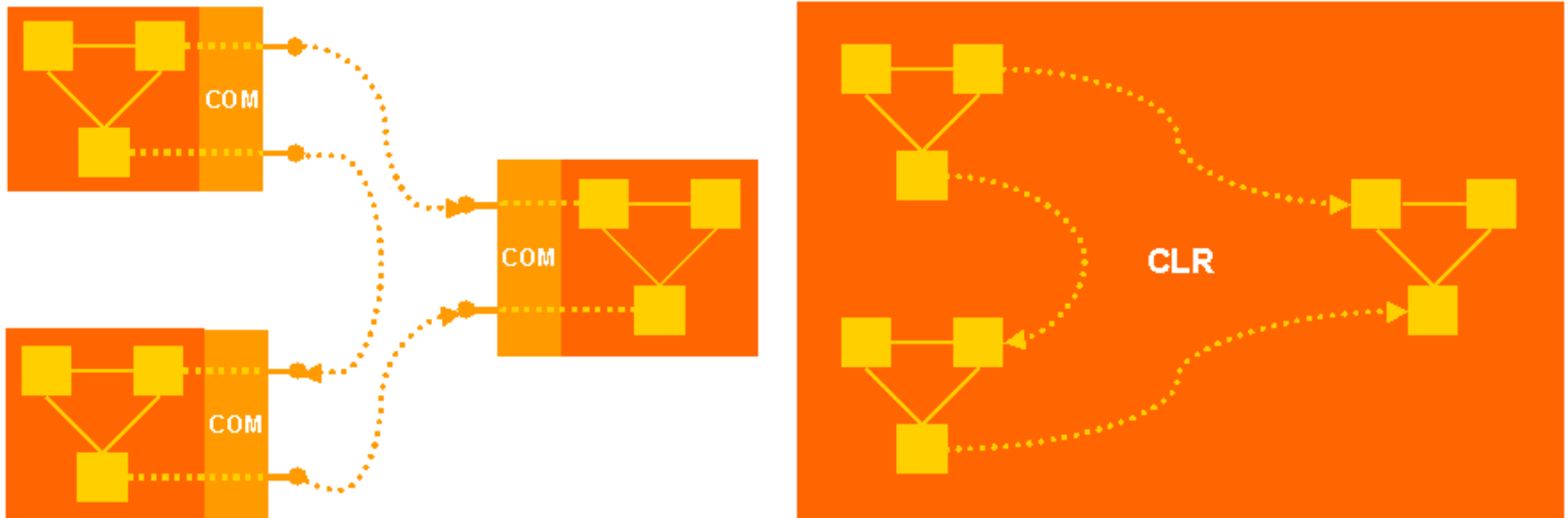




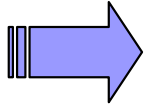
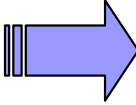
Common Type System CTS (1)

- alle .NET Programmiersprachen implementieren dasselbe Typsystem
- „jedes Teammitglied programmiert in der Sprache seiner Wahl“
- Interoperabilität zur Laufzeit und zur Entwicklungszeit!

Common Type System CTS (2)



Die Säulen von .NET

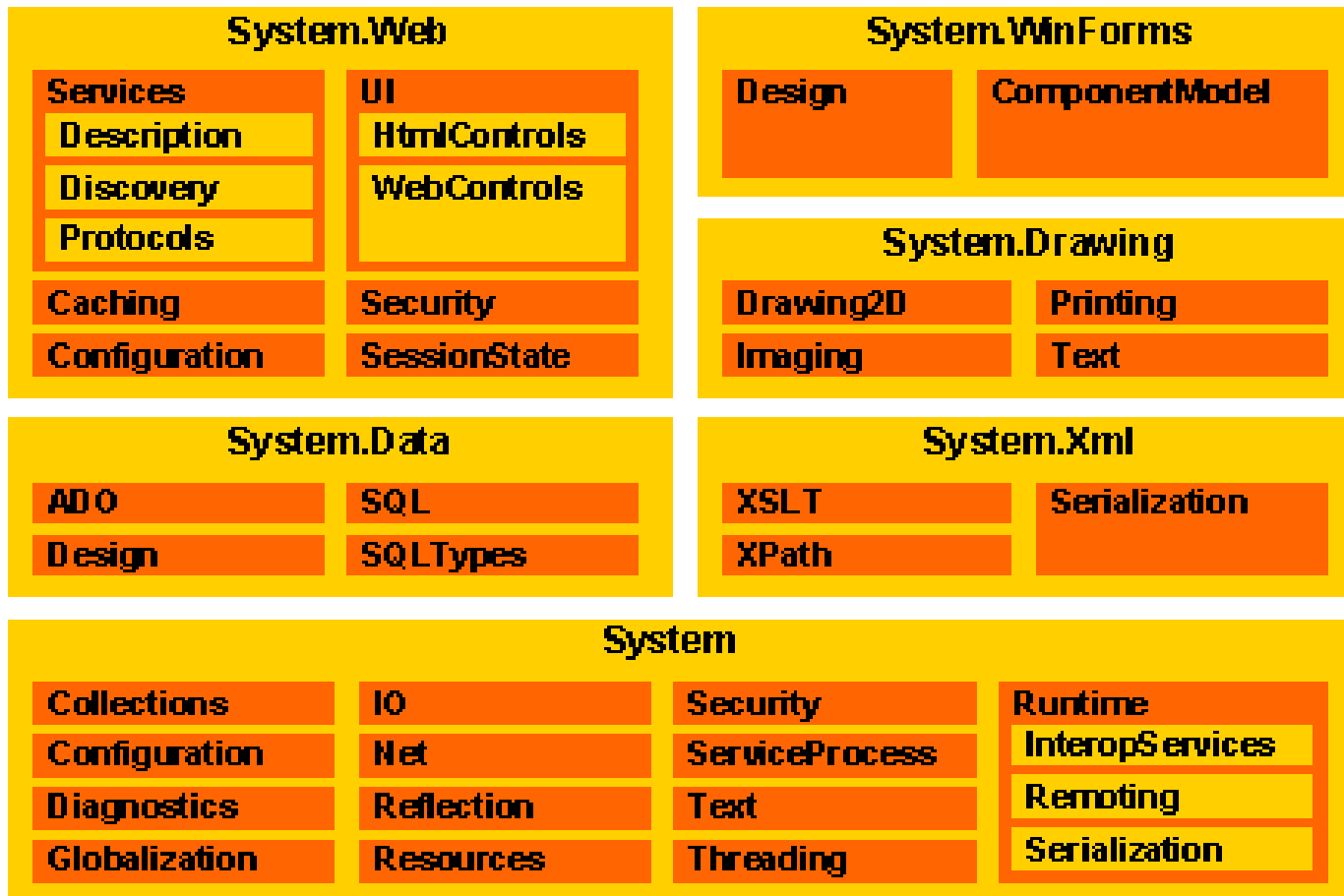
- Programmierplattform .NET Framework 
- .NET Enterprise Server 
- .NET My Services: Webservices für Standardprobleme (†)
- .NET Devices (Mobile Internet Toolkit, .NET Compact Framework)



.NET Framework (1)

- Entwicklungs- und Laufzeitplattform
- Common Language Runtime
- Compiler für VB, C#, C++
- Klassenbibliothek

.NET Framework (2)



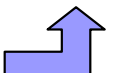


.NET Enterprise Server (1)

- SQL Server
 - RDBMS
- Exchange Server
 - Messaging & Collaboration
- BizTalk Server
 - Weiterleitung und Verteilung von Nachrichten und Dokumenten innerhalb von und zwischen Unternehmungen

.NET Enterprise Server (2)

- SharePoint Portal Server
 - Dokumentenmanagement
- Commerce Server
 - Framework für e-Business Lösungen
- ...



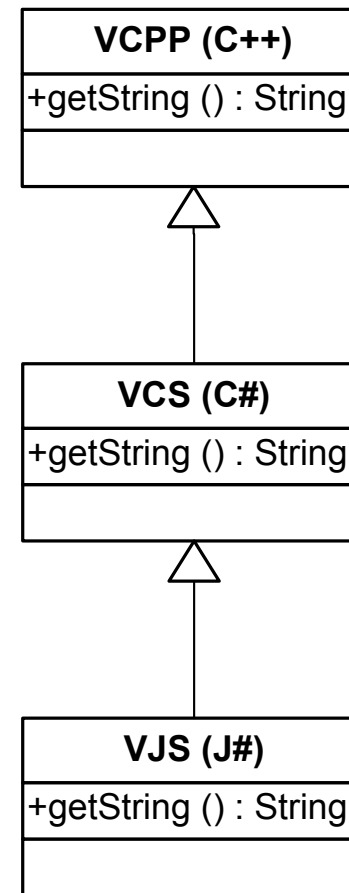


.NET Framework 2.0 / VisualStudio 2005

- Generics
- Partial Class Definitions
- Debugger Edit & Continue Support

Beispiel

- Vererbung über Sprachgrenzen hinweg



Referenzen & Links (1)

- Microsoft .NET Framework Home
<http://msdn.microsoft.com/netframework/>
- GotDotNet .NET Framework Community
<http://www.gotdotnet.com/>
- CodeZone.ch
<http://www.00001001.ch/>
<http://www.codezone.ch/>



Referenzen & Links (2)

- Westphal Ralf: .NET kompakt, Spektrum Akad. Verlag, Heidelberg 2002