

---

# INTERSYSTEMS

# SYMPOSIUM 2005

Nové jazykové brány  
do Caché



Daniel Kutáč

---

# O čem budeme mluvit



- **.NET**
- **T/SQL**
- Perl
- Python
- MultiValue Basic

**Téma**



**.NET provider**

# .NET Provider

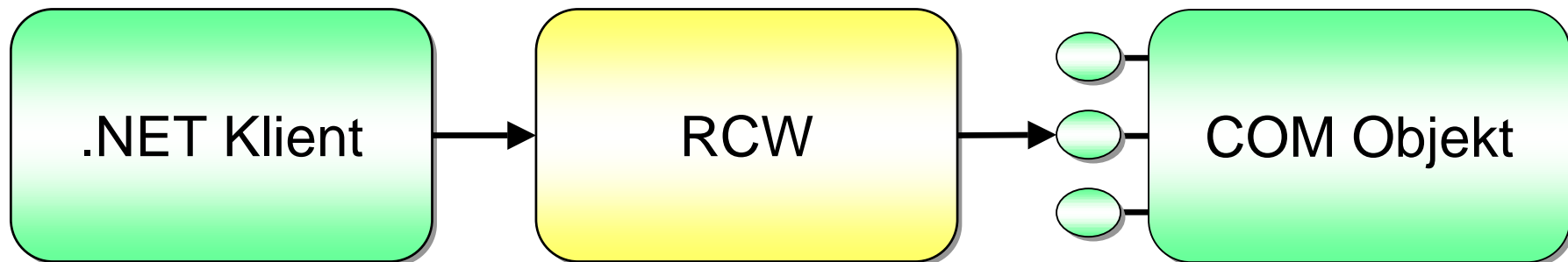


- Co lze již dnes
  - Factory / VisM
  - ODBC.NET
  - Web Services

# Factory a VisM



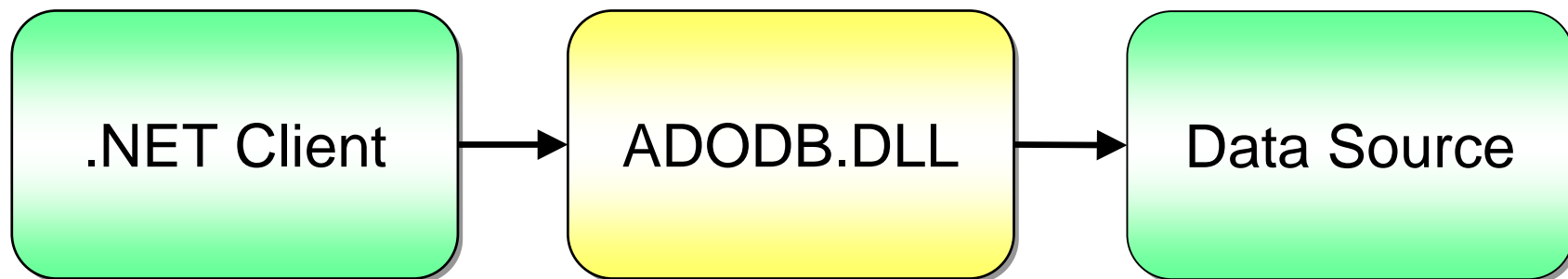
- Zapouzdří se do .NET obálky pomocí mechanismu RCW
- COM objekty se tváří jako nativní .NET objekty a naopak



# ODBC.NET



- Samostatná komponenta .NET frameworku
- Kompatibilní s ADO 2.7



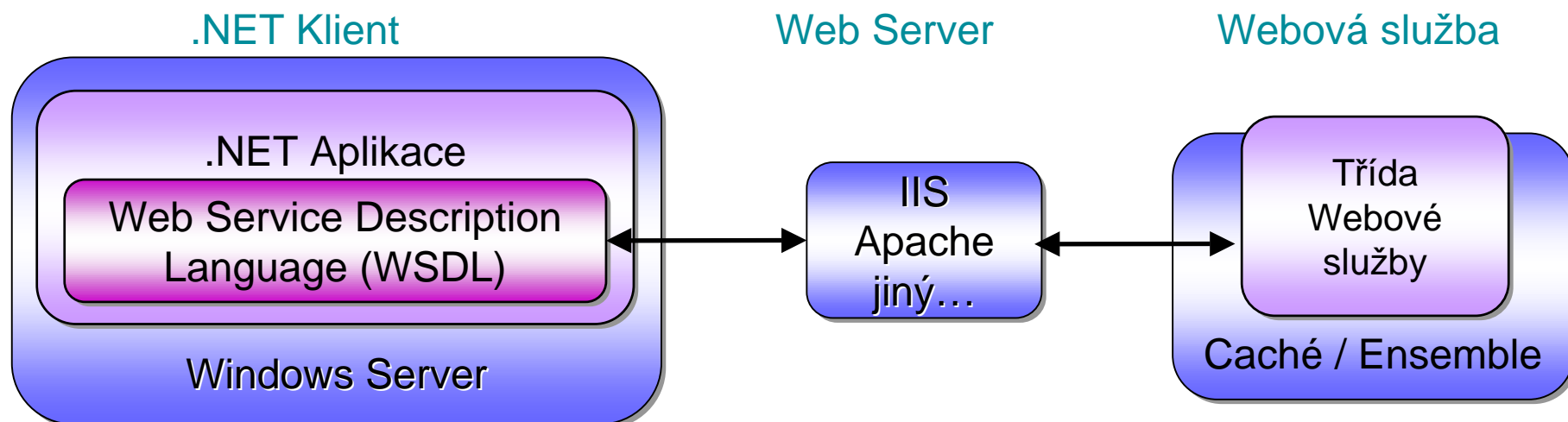
# Webové Služby



- Na rozdíl od COM a ODBC.NET jenž jsou v .NET pouze kvůli kompatibilitě „zděděných“ aplikací jsou Webové služby v .NET novou technologií a je na ně kladen důraz.
- Caché umí vystavovat webové služby velmi přirozeným způsobem s minimem práce

# Webové Služby II.

- **Al**: HTTP protokol -> bezstavová komunikace
- Metody POST a GET



# .NET Framework



- Obecná objektově orientovaná základna pro mnoho jazyků
- Jednodušší model pro nasazení aplikací
- Automatizovaná správa paměti
- Standardizovaný způsob řešení konfliktů softwarových verzí
- Bohatší bezpečnostní model
- Jednoduchý způsob uspořádání systémového a uživatelského kódu do jmenných prostorů
- “Téměř dokonalá” zaměnitelnost s COM komponentami

# Řízený a neřízený kód



- Řízený kód (MSIL)
  - Běží v .NET Common Language Runtime
  - Těží z nových vlastností .NET, jako např. garbage kolektoru, verzování kódu a bezpečnostních politik
  - Runtime „nezávislý na platformě“
- Neřízený kód
  - Předkompilován pouze pro specifickou architekturu
  - Běží pouze na dané platformě
  - Náchylnější na různé chyby ve správě paměti, verzování a bezpečnostní rizika

# ADO.NET

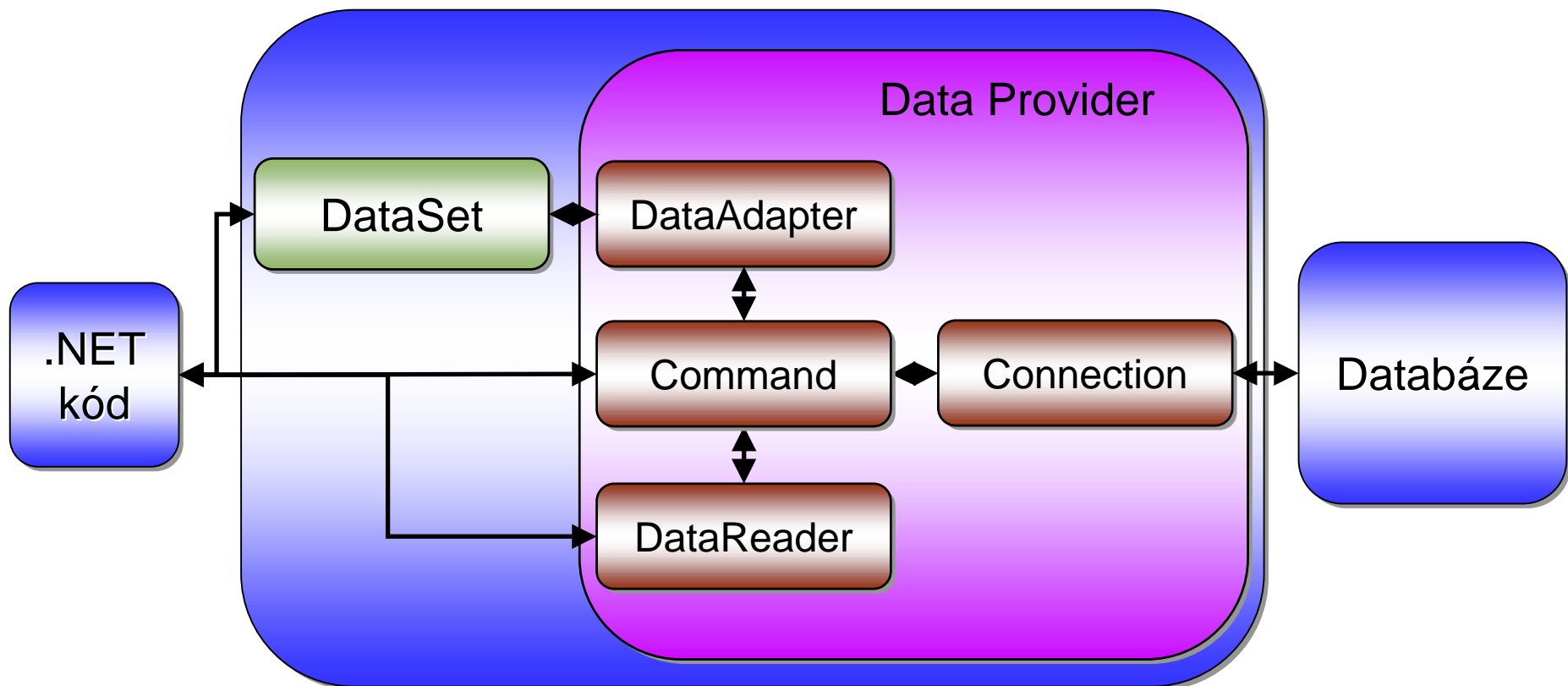


- Poskytuje propojení mezi .NET programem a datovým uložištěm
- Konzistentní „objektově orientovaný“ pohled na data nezávisle na typu (databáze)
- Dodavatelé databází mohou implementovat ADO.NET různými způsoby

# ADO.NET - architektura



DB Server



# Caché + Řízený kód



- Výhody propojení s Caché pomocí Managed Code
  - Automatické generování .NET tříd přímo z Caché tříd
  - Skutečné objekty na všech úrovních aplikace
  - Běží se všemi výhodami řízeného kódu
  - Sdílí některé komponenty s ADO.NET

# Generování .NET tříd z Caché tříd



- Knihovna ClassBinder
  - Může být volána samostatně či jako externí nástroj Visual Studia
  - Texty formulářů mohou být snadno lokalizovány
  - Vygenerovaný kód je přímo použitelný v .NET projektech
  - Generuje buď zdrojový kód nebo sestavený kód

# .NET Provider



- Ukázka vygenerování jednoduchého formuláře



**Téma**



**T/SQL**

# T/SQL



- T/SQL je sada programových rozšíření SQL od firem Sybase a Microsoft.
- Zahrnuje mj.
  - Transakce
  - Dočasné objekty
  - Podmíněné zpracování
  - Zpracování výjimek
  - Deklaraci proměnných
  - Práci s řádky záznamů apod.

# T/SQL



**Stored Procedure Properties - SalesByCategory**

General

Name: SalesByCategory

Owner: dbo

Create date: 6.8.2000 1:34:53

Text:

```
CREATE PROCEDURE SalesByCategory
  @CategoryName nvarchar(15), @OrdYear nvarchar(4) = '1998'
AS
IF @OrdYear != '1996' AND @OrdYear != '1997' AND @OrdYear != '1998'
BEGIN
  SELECT @OrdYear = '1998'
END

SELECT ProductName,
  TotalPurchase=ROUND(SUM(CONVERT(decimal(14,2), OD.Quantity*(1-OD.Disc
FROM [Order Details] OD, Orders O, Products P, Categories C
WHERE OD.OrderID = O.OrderID
  AND OD.ProductID = P.ProductID
  AND P.CategoryID = C.CategoryID
  AND C.CategoryName = @CategoryName
  AND SUBSTRING(CONVERT(nvarchar(22), O.OrderDate, 111), 1, 4) = @OrdYear
GROUP BY ProductName
ORDER BY ProductName

GO
```

21, 18/21

- Ukázka T/SQL
- Uložená procedura

# T/SQL



- Přístup Caché k T/SQL
  - Metody tříd, uložené procedury
  - Language = tsql
  - Kompilace – ObjectScript nebo zůstává T/SQL
- TSQL Shell
  - Do ^%apiTSQL
- Dostupné jako AdHoc release

# T/SQL



- Co je k dispozici
  - Query X() as %TSQL.Query [ SqlProc ]
  - ClassMethod X() as ... [ Language = tsql, SqlProc ]
  - CACHE { ... kód v ObjectScriptu }

Ukázka!



- Co je potřeba udělat ručně
  - Triggery
    - Založeny na datových setech, v Caché na řádcích => nutno přepsat
  - Pojmenované transakce



- Průvodce migrací uložených procedur \*
- Pro každou uloženou proceduru vytvoří samostatnou třídu s kódem v ObjectScriptu

\* Caché 5.1

**Téma**



# **Perl a Python**

# Perl a Python



- Perl
  - Practical **E**xtraction and **R**eport **L**anguage
  - Optimalizovaný pro práci s řetězci, I/O a systémovými úlohami na Unixových OS (běží i pod Windows)
  - Inspirován C, C++ a Fortranem
  - S rozvojem WWW se stal populárním pro psaní CGI programů

# Perl a Python



- Python
  - Podobně jako Perl je skriptovacím jazykem
  - Na rozdíl od Perlu je v něm možno psát i obecnější programy
  - Portován pro různé OS
  - Má rozhraní do různých systémů správy oken (X11, MFC, Mac, ....)

**Téma**



# **MultiValue Basic**

# MVBasic



- MultiValue
  - Technologie rozšířená hlavně v USA
  - Podobně jako M technologie založena na vícerozměrném datovém modelu.
  - Hlavní implementace – PICK, Unidata, UniVerse

# MVBasic



- Caché nabízí nový typ dokumentu
- MVB dokumenty se podobají MAC dokumentům
- Jsou psány v jazyce MVBasic
- Kompilují se do stejného objektového kódu jako MAC/INT rutiny
- Zaměřeno hlavně na migrace komunity MultiValue

Ukázka!

---

# INTERSYSTEMS

# SYMPOSIUM 2005

Nové jazykové brány  
do Caché



Děkuji!

---