

.NET プラットフォーム概説

マイクロソフト株式会社
デベロッパーマーケティング本部
.NET テクノロジー部
野村 一行

Microsoft
.net™

XML: 次世代ソフトウェアの 共通言語 (Lingua Franca)

XML

Web

GUI

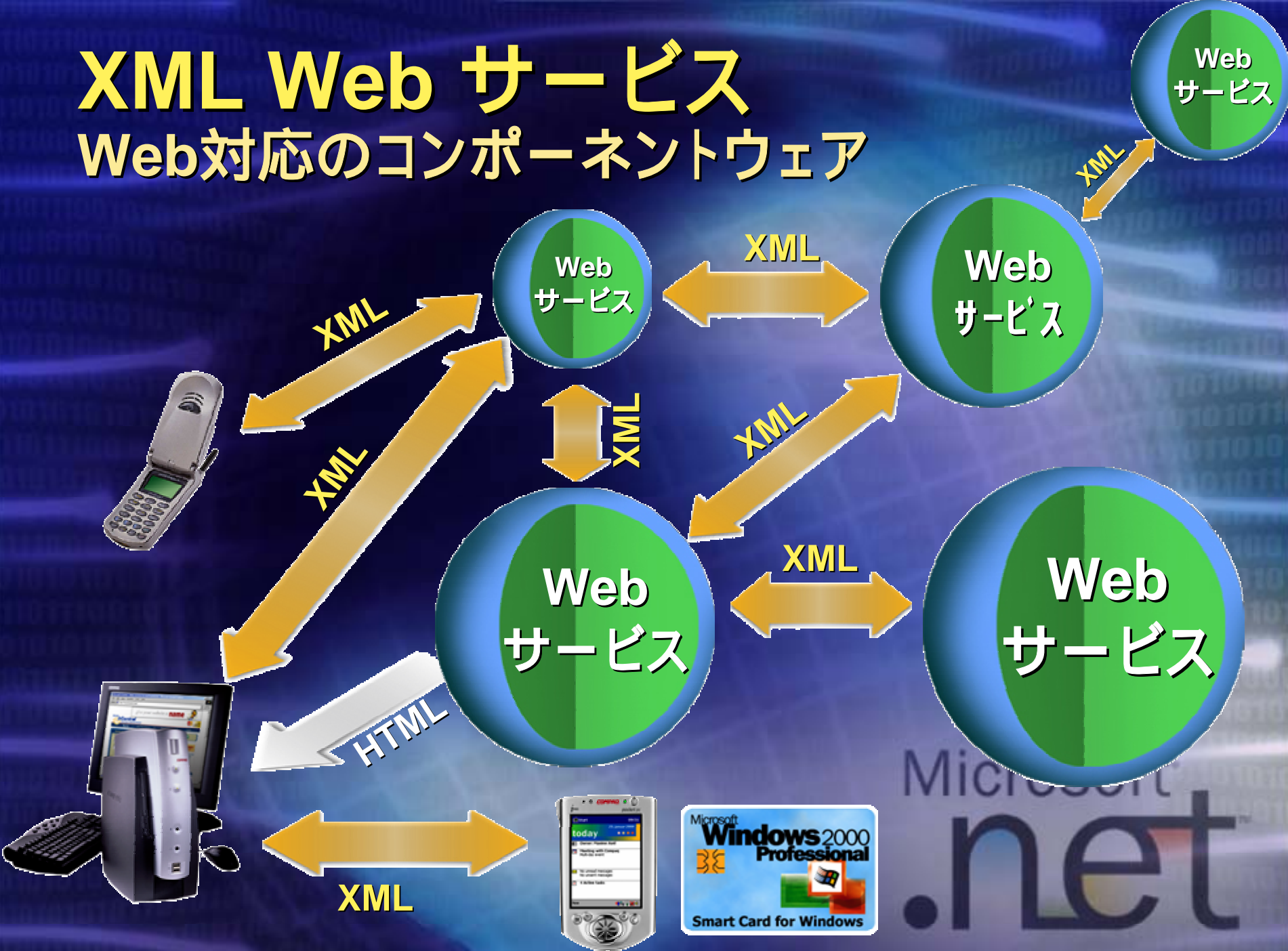
PC

オープンソフトウェアの 新世代プロトコル

- 統一データフォーマット
- 標準技術を利用した統合
- 双方向の適応性
 - ユーザー(デバイス)からの適応性
 - コンピュータ間の通信適応性
 - 柔軟なカスタマイズ
- プログラム可能な XML
Web サービスの出現

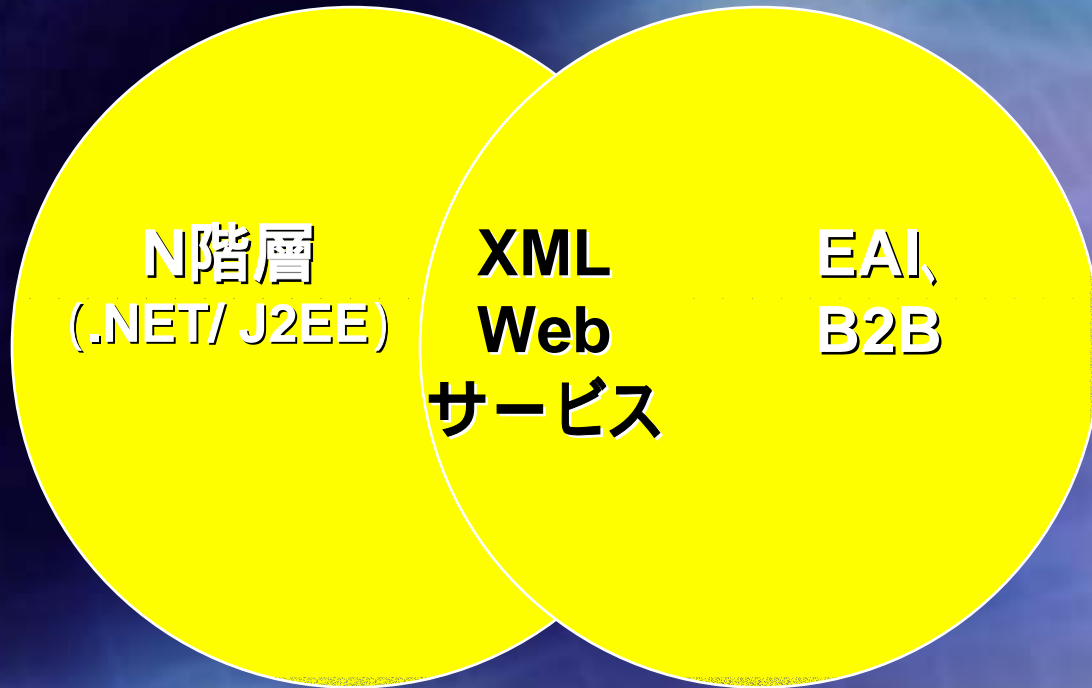
XML Web サービス

Web対応のコンポーネントウェア



XML Web サービスの2つの視点

Web アプリケーション アプリケーション統合



- N階層アプリケーションのWeb対応
- Web技術を用いたアプリケーション統合アプローチ



XML Web サービスへの パラダイムシフト

中央集権型
コスト削減

ベンダ中心/抱え込み
ベストプラクティス/固定化
シンククライアント

2相コミットランザクション
単一プラットフォーム

アプリケーションの垂直/階層統合
プラットフォーム独自セキュリティ

レコード、オブジェクトの形によるデータ交換
ドメイン内ディレクトリからアプリケーション発見

画面遷移フレームワーク
オブジェクトによるルーティング

実装の再利用
物理ノードのロードバランシング
スケーラビリティ・応答性

自律・分散・協調型
価値増大・収穫逓増

ユーザー中心/戦略的アウトソーシング
俊敏性/流動性

リッチクライアント

マルチ/ロングランザクション
マルチプラットフォーム

アプリケーションの水平/ハブ&スポーク統合
インターネット標準セキュリティ

XMLによるデータ交換

インターネットワイドなディレクトリからサービス発見
ビジネスプロセスフレームワーク

メッセージ内容によるルーティング
サービス(機能)の再利用

Web サイト間のロードバランシング
可用性・信頼性・管理容易性

XML Web サービスの構成要素

オープンな
インターネット
プロトコル



XML Web
サービス

標準 Web プロトコルを介してアクセスする、
プログラム可能なアプリケーション
コンポーネント

- インターネット上のディレクトリサービス
- Web サービスを利用するためのメタデータ情報 (IDL に相当)
- Web サービスを利用するためのメッセージ内容 (XML ベース)
- インターネット標準プロトコル

UDDI

*Universal Description,
Discovery, and Integration*

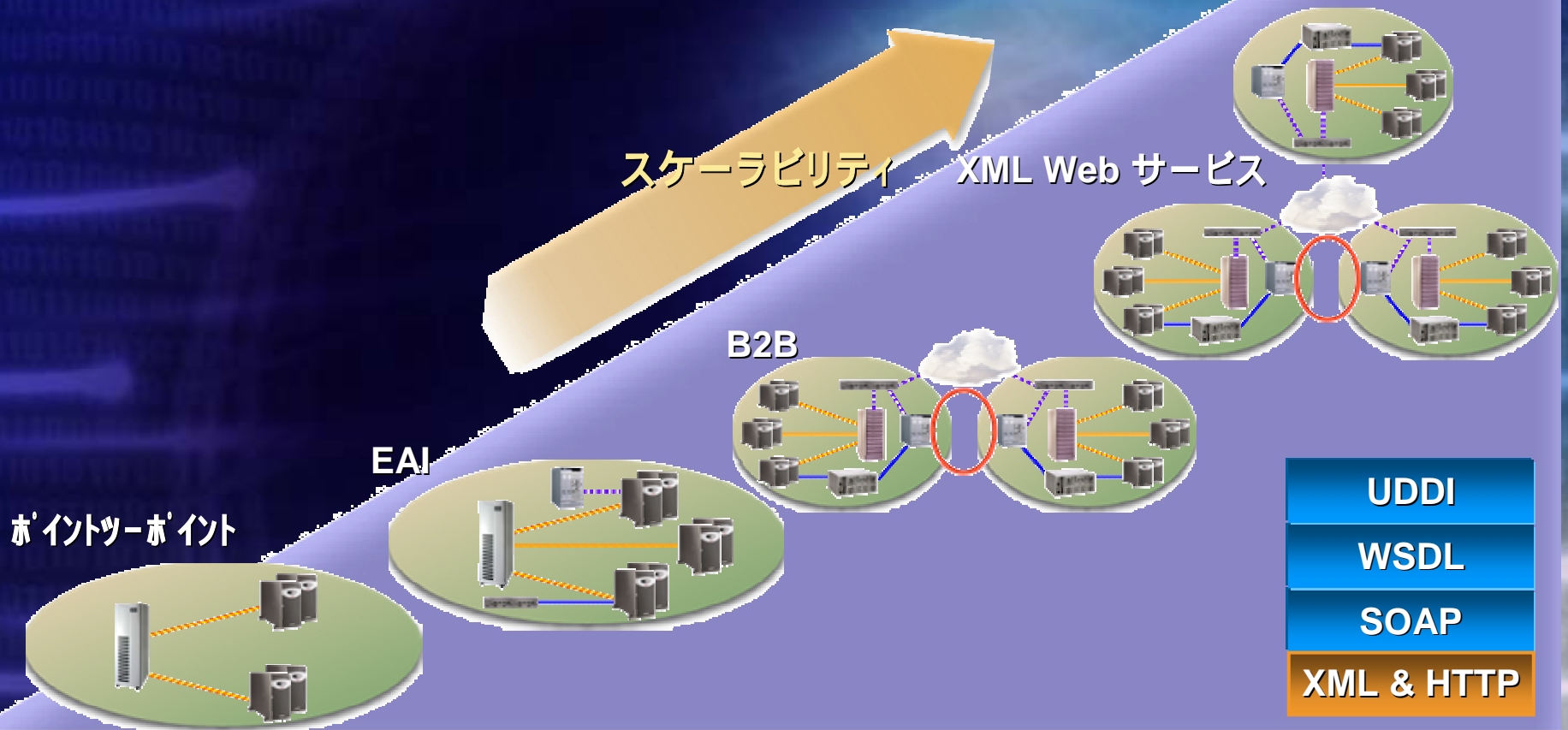
WSDL

*Web Services
Description Language*

SOAP

XML & HTTP

XML Web サービスへの連続性



疎結合性

.net™

Microsoft .NET

XML Web サービス プラットフォーム

- クライアント、サーバー、サービス
- 一貫したプログラミングモデル、画期的な生産性向上を実現したツール
- XML Web サービスによる相互運用性
- 主な特長
 - サービスとしてソフトウェアを公開・配布
 - ソフトウェア統合を前提に設計
 - サービスを意識したシステム構築を可能に
 - 新しく豊かなユーザー体験

Microsoft
.net™

Microsoft .NET



.NET 開発者へ提供する技術群

Visual Studio .NET

使いやすい、ドラッグ&ドロップ
ベースのビジュアル設計・開発環境

.NET Framework

疎結合アーキテクチャを
取り入れた、共通インフラ

Windows
アプリケーション基盤

ハイパフォーマンス アプリケーショ
ンのための共通基盤 (Windows、
COM+、サーバー群)

通信プロトコル
(XML、SOAP、HTTP、HTML)

オープンなインターネット標準

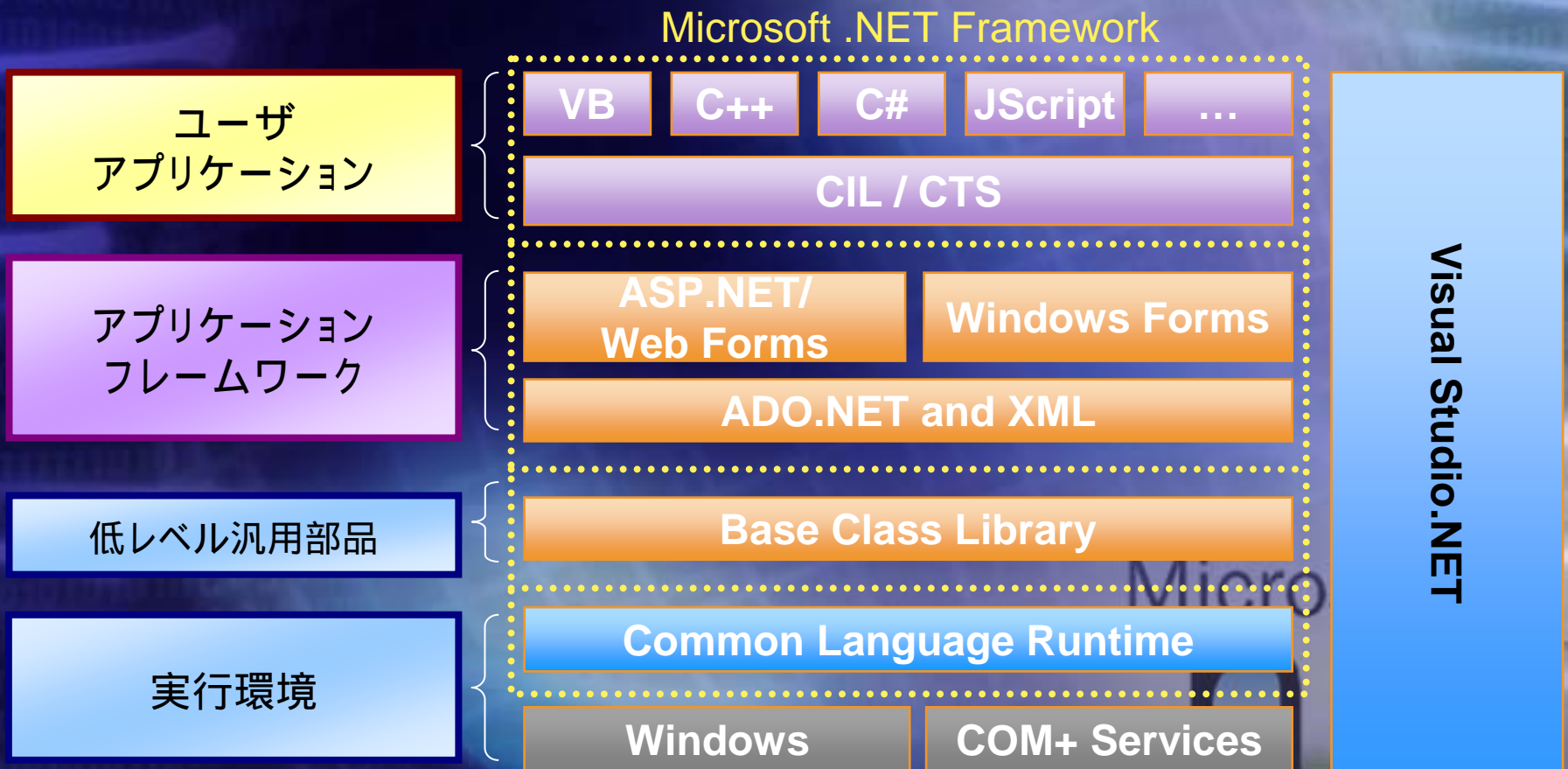
.NET My Services

ユーザー中心の XML
Web サービス

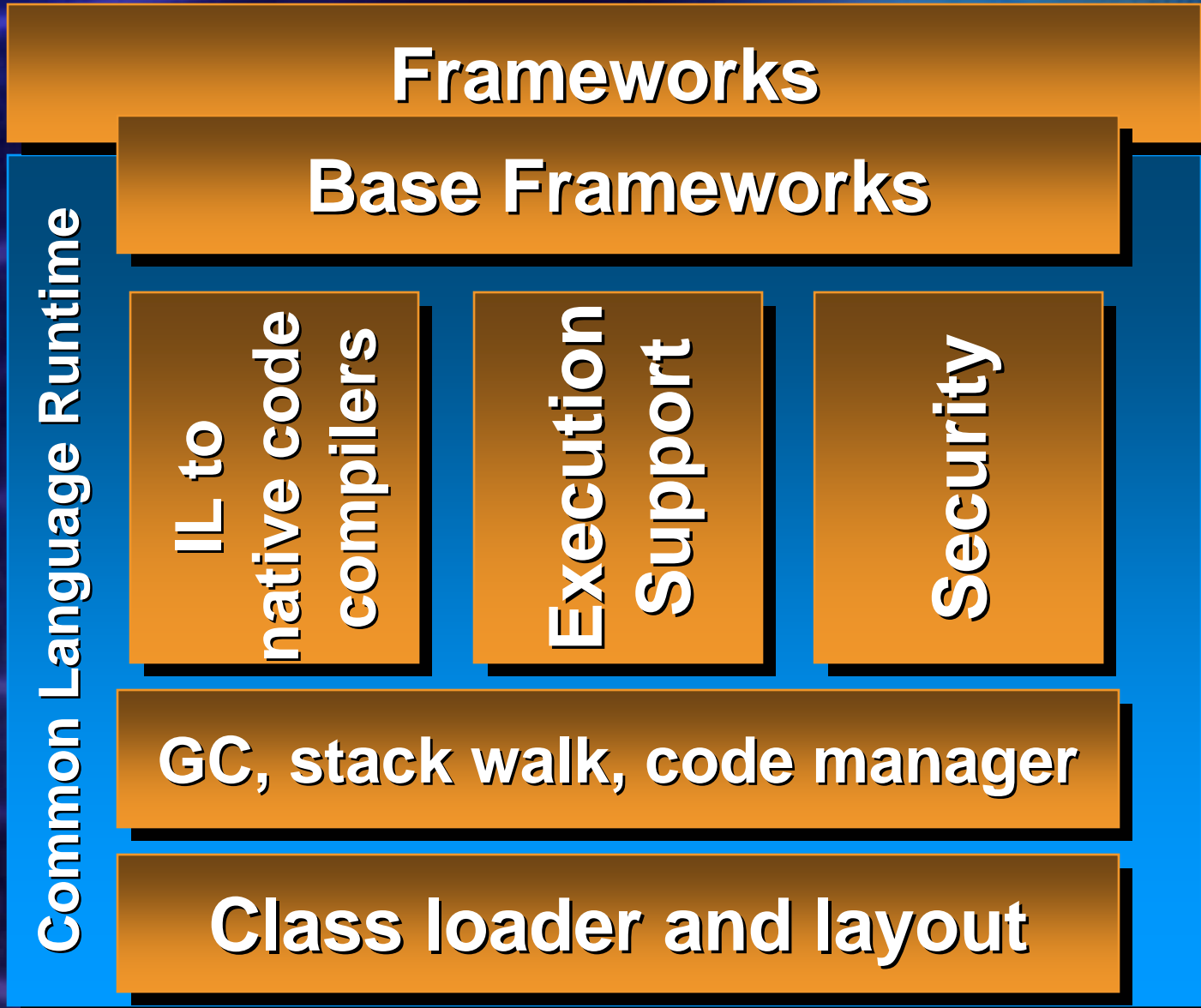
Microsoft
net™

.NET Framework の構成図

- XML Web サービスまで包含した、一貫したプログラミングモデル

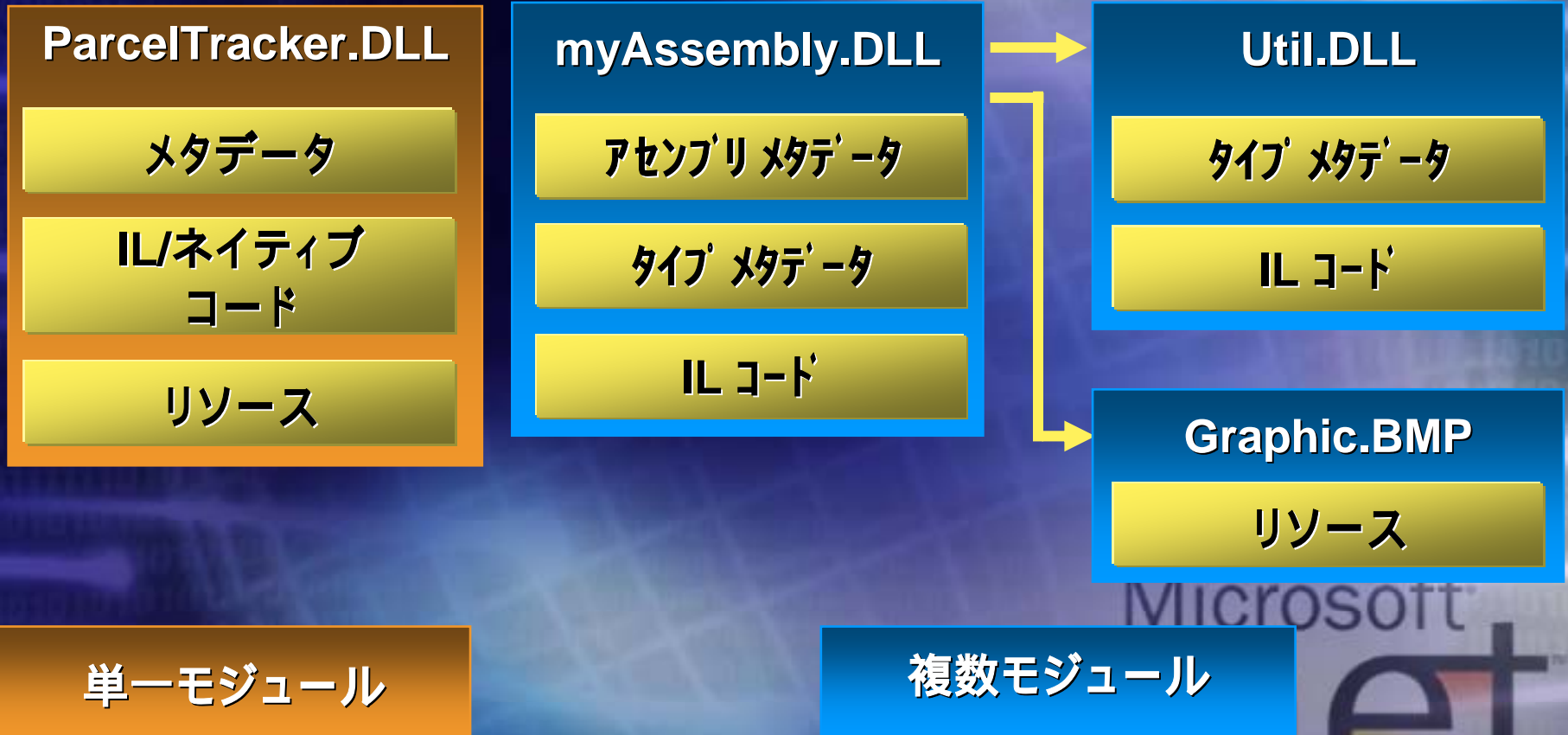


共通言語ランタイム

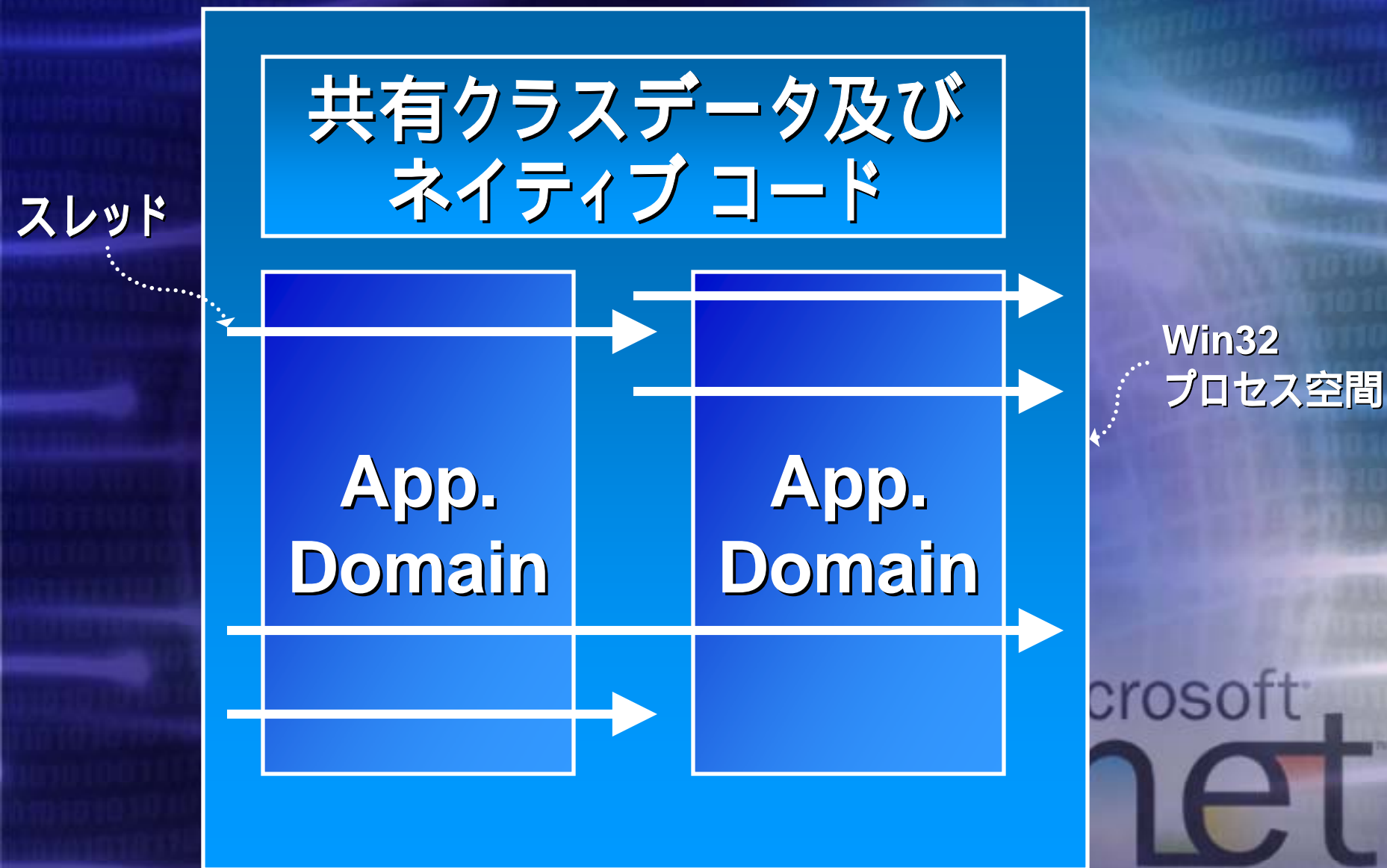


アセンブリ

論理的なパッケージ: 配布管理の単位



実行時のプロセスモデル



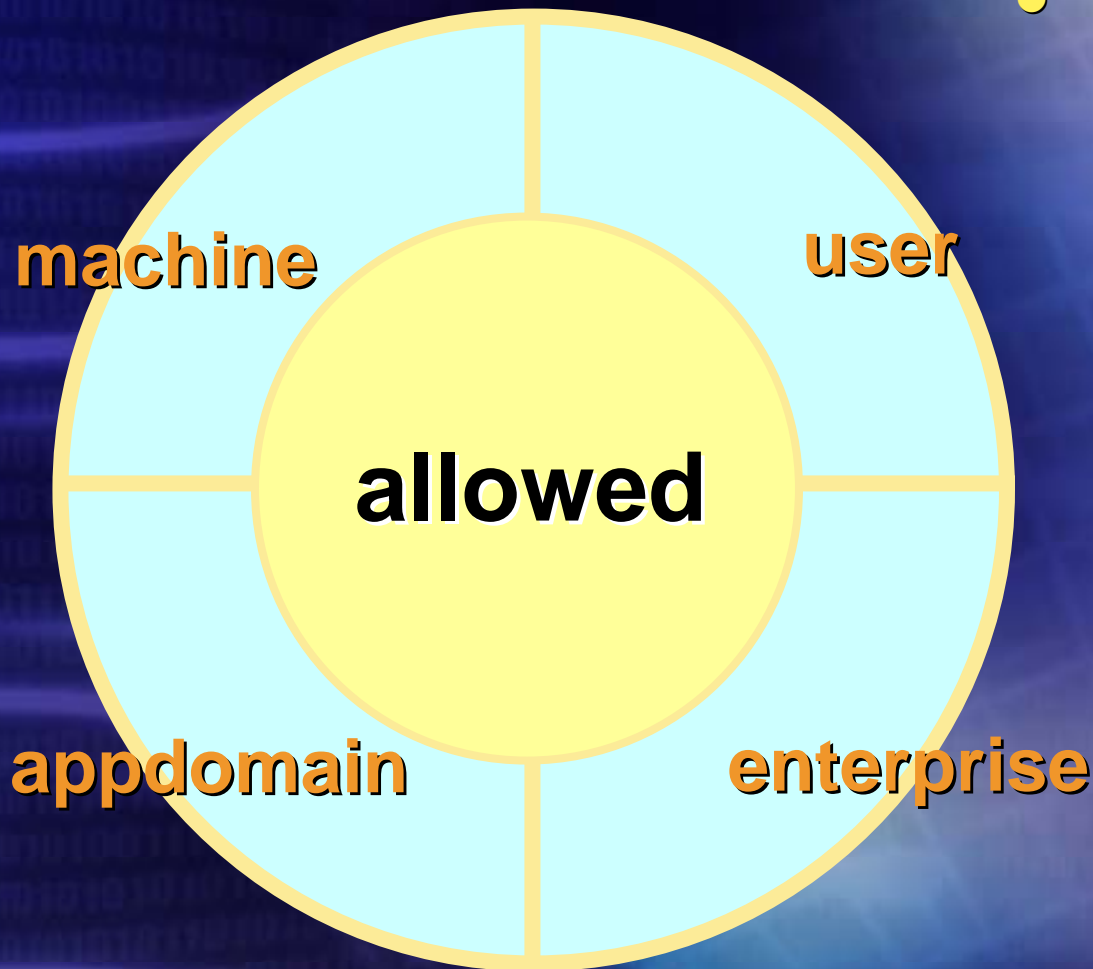
エビデンスベースのセキュリティ

全体動作



セキュリティ ポリシー

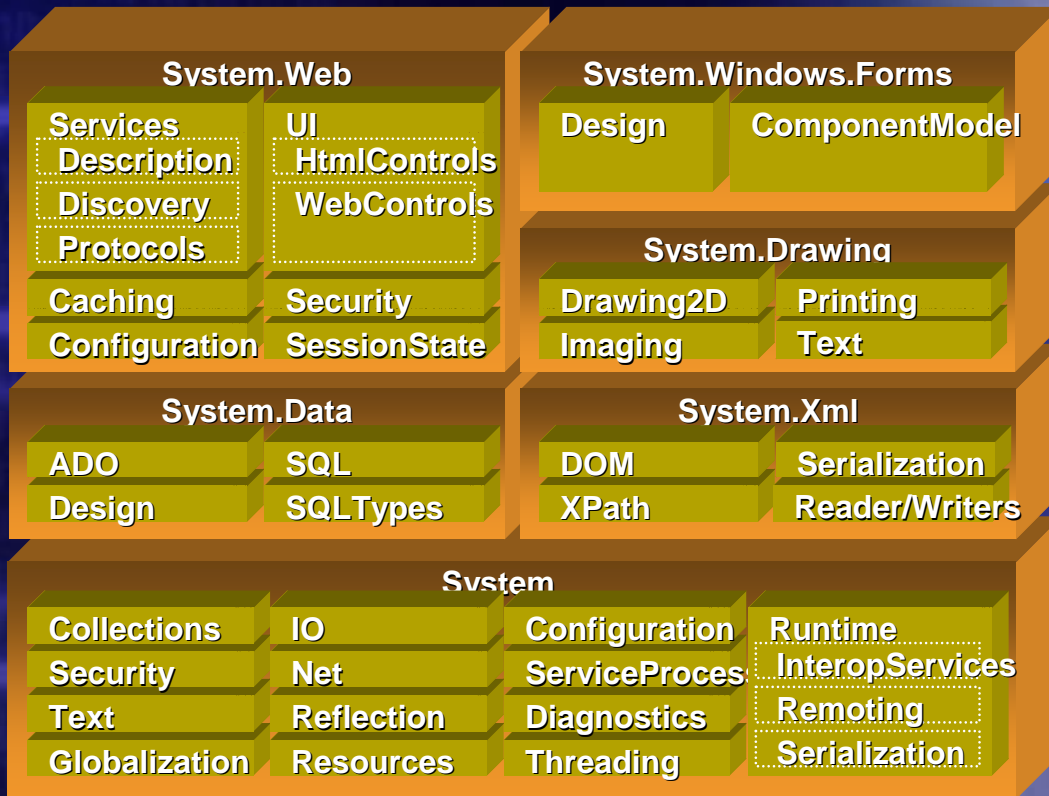
異なったポリシーレベルの論理積(AND)



- 複数レベルのポリシー
 - デフォルトでは、マシン、ユーザ、エンタープライズの単位
 - マシン、ユーザーおよびエンタープライズレベルの構成:
 - caspol.exe
 - ConfigWizards.exe
 - Mscorcfg.msc
 - より詳細なポリシー制限は、アプリケーションドメイン単位で可能
 - アプリケーションドメイン構成は、基本クラスライブラリのAPI呼び出し

.net

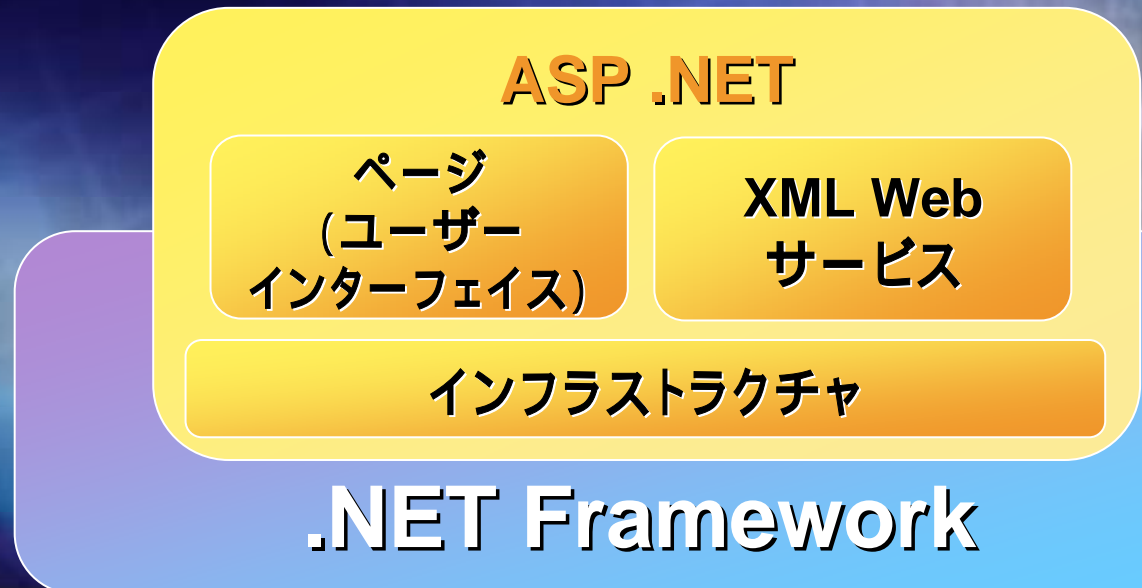
.NET Framework の主なクラス群



Microsoft
.net

ASP.NET 概要

- .NET における Web ソリューションの実行環境として
 - ユーザーインターフェイスの生成機能
 - XML Web サービスの構築機能
 - アプリケーション環境を支えるインフラ技術の提供
- .NET Framework の恩恵を享受



ASP.NET – 3つの役割 (1/3)

ユーザーインターフェイスの生成

- Web フォームによる自動クライアントサポート
 - Web UI のためのコントロールライブラリ
 - Web フォーム
 - モバイル Web フォーム
 - デバイスに応じた UI を自動的にレンダリング
 - 共通化されたプログラミングモデル
 - サーバー側イベントドリブン プログラミング モデルを採用し、アプリケーションをデバイス非依存に
- コードとコンテンツの分離
- コンパイル可能な言語のサポート
 - Visual Basic .NET、Visual C# .NET、JScript
 - COBOL、Perl など

Microsoft
.net

ASP.NET – 3つの役割 (2/3)

XML Web サービスの実現

- .asmx ファイルによる XML Web サービスの構成
 - ?WSDL パラメータを与え、WSDL 情報を自動生成
 - HTML を利用した情報ページの自動生成
 - 単純なサービスのテストの実行
 - 送受信されるメッセージやその書式についても情報ページで提供
- 開発容易なプログラミングモデル
 - 開発手法に変更なし
 - WebMethod 属性の付加により Web サービスとして構成
- XML Web サービスの利用
 - SOAP / HTTP-POST / HTTP-GET を利用した呼び出し
 - サービス利用時にこれらの手段を意識する必要なし

Microsoft

net

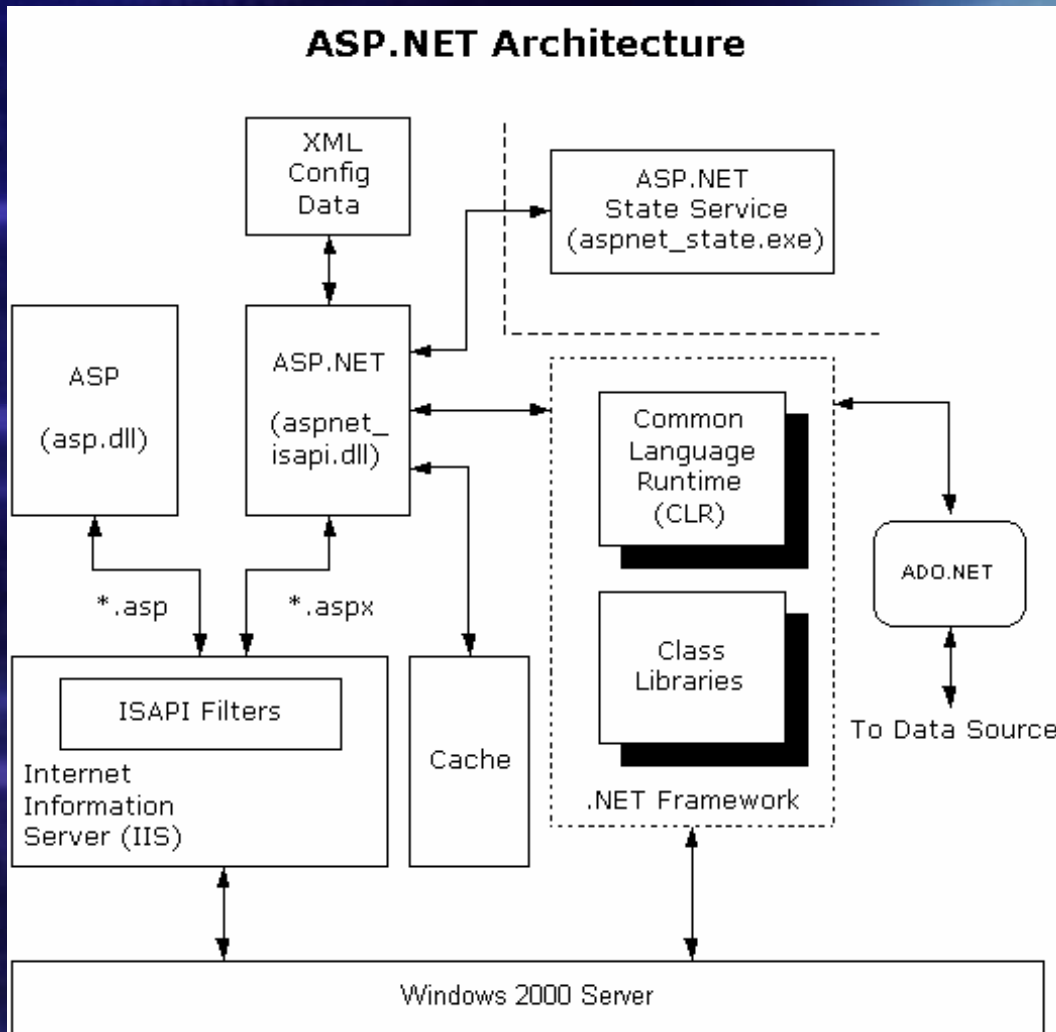
ASP.NET – 3つの役割 (3/3)

Web ソリューションのインフラ技術の提供

- アプリケーション設定情報をファイルで保持
 - レジストリや IIS メタベースなどを使用しない
 - XML ベースの設定ファイル設定情報を保持
 - 変更されると即時に Web アプリケーションへ反映
- 動的なコード更新
 - DLL ロックやリブート、サービス再起動からの開放
 - 容易なアプリケーション配布を実現
 - XCOPY ライクなモジュール配置
- キャッシング機構の提供
 - Output Cache, Fragment Cache
 - Cache API

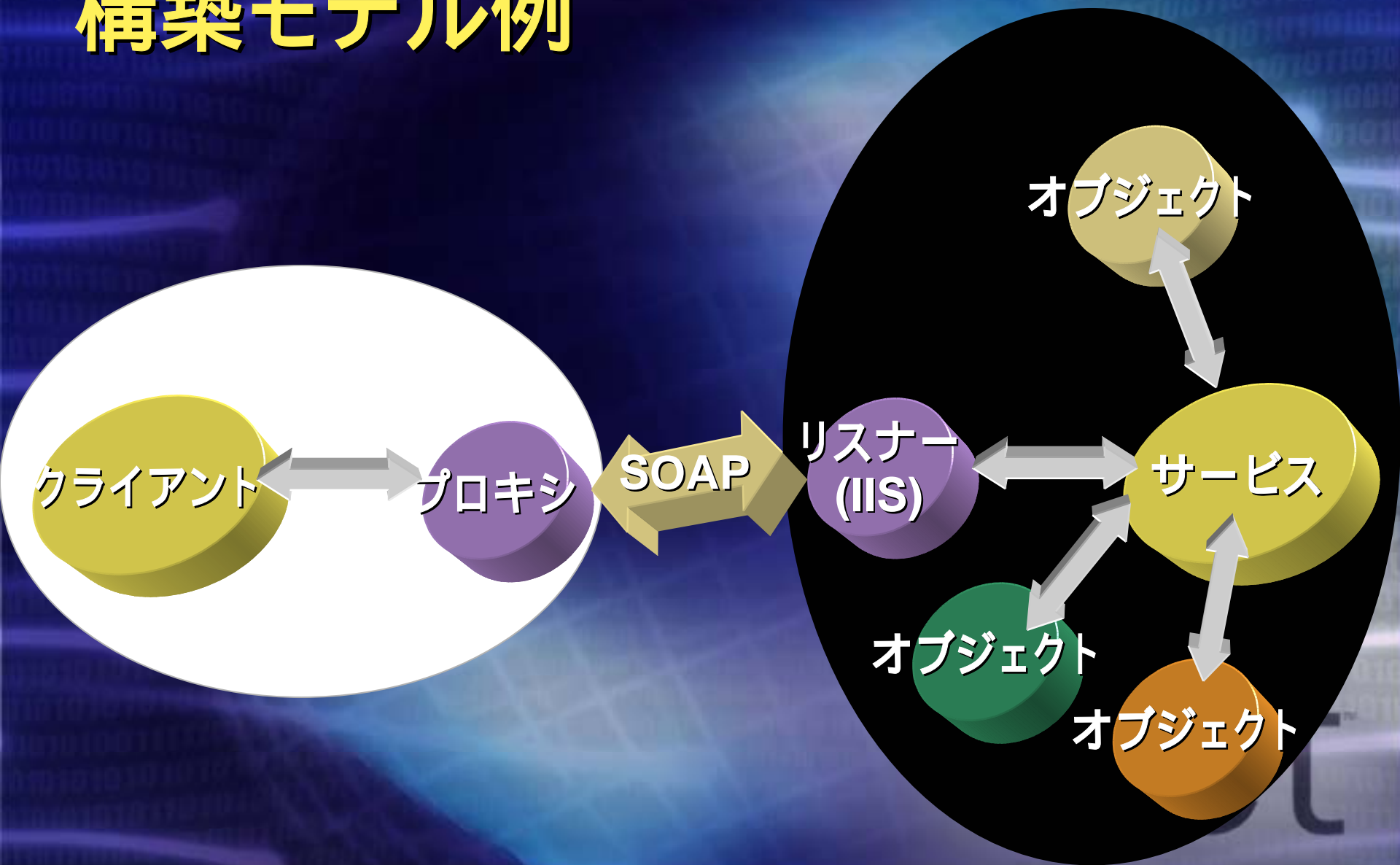
Microsoft
.net™

ASP.NET アーキテクチャ



Microsoft
.net

XML Webサービスの構築モデル例



XML Web サービスの開発

シンプルな実装方法を実現

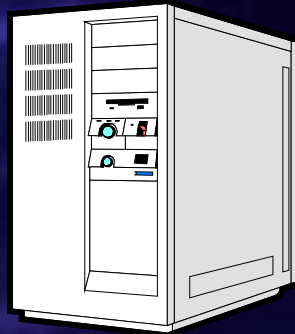
- Visual C# .NET における XML Web サービスの実装
 - *FileName.asmx* ファイルとして作成
 - **[WebMethod]** 属性の付与によって実装

```
<%@ WebService Language="C#" %>
using System;
using System.Web.Services;
public class clsHello : WebService{
    [WebMethod]
    public String HelloWorld(){
        return "Hello World";
    }
}
```

Microsoft
.net

サービス要求のための プロキシ生成

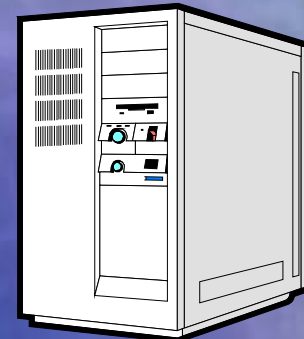
スキーマコンパイラ
(wsdl.exe)



TrackOrder.asmx?WSDL



サービス記述情報(XML)



プロキシ
クラス

コードコンパイル

プロキシ
DLL

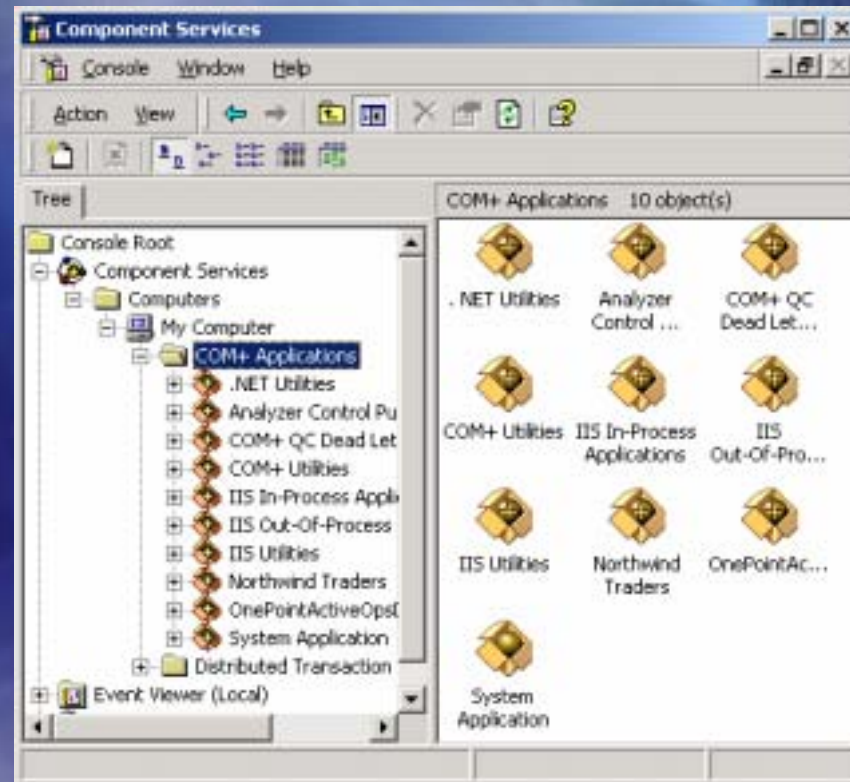


.ASMX

Micr
.net

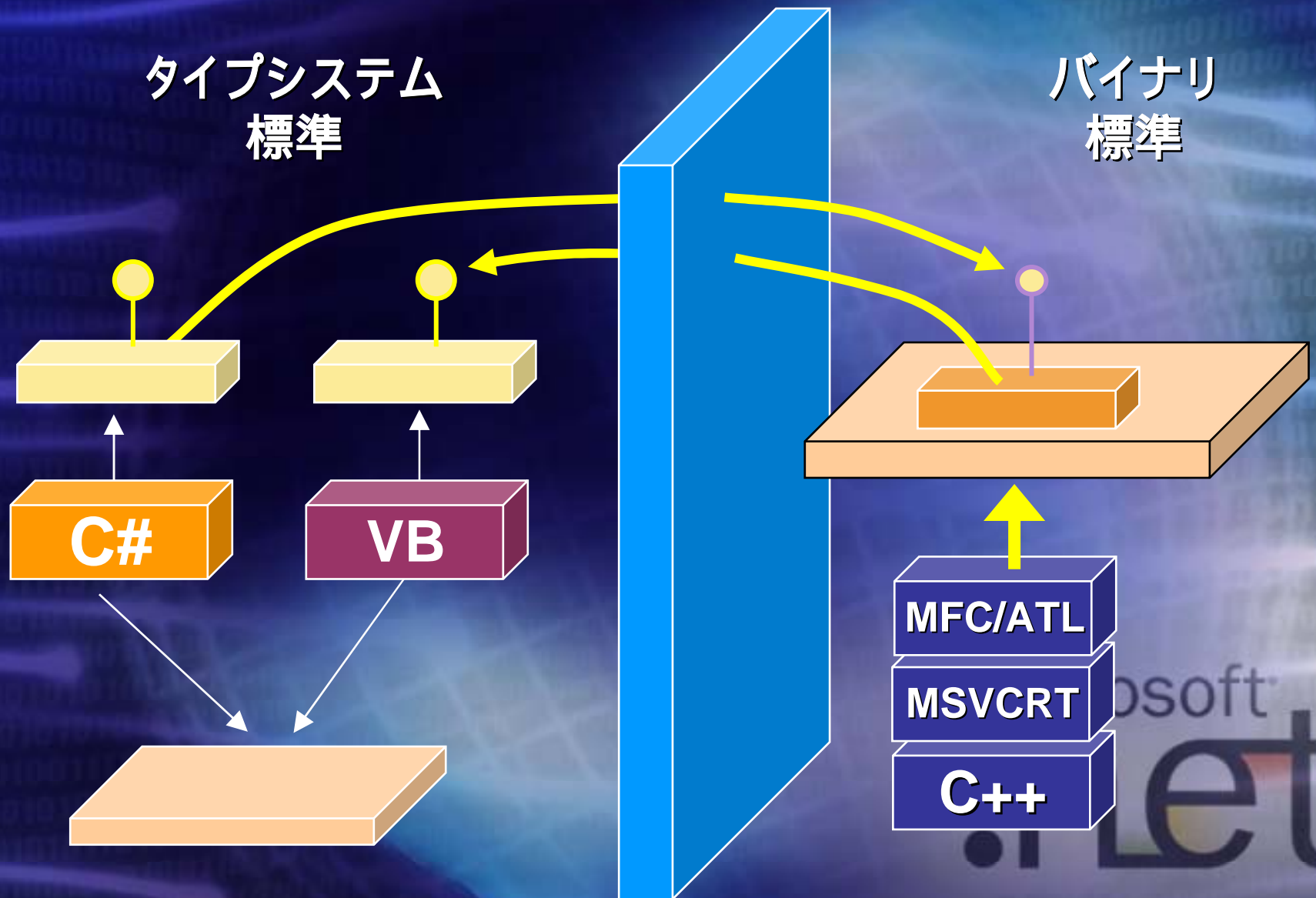
.NETとCOM+ サービス

- .NET Framework と統合
 - System.EnterpriseServices
 - ServicedComponent クラス
 - 属性によりサービスを定義
- Windows 2000以降のCOM+ サービスが全て利用可能
 - トランザクション、オブジェクトプーリング、キューコンポーネント、ロールベースのセキュリティ、疎結合イベント、BYOT、Compensating Resource Manager、...



.net

COM 相互運用サービス



他プラットフォーム間連携

- Soap Builders

- 主要製品におけるSOAPベースの相互運用性を検証
- SoapBuilders Interoperability Lab を参照
 - <http://www.xmethods.net/ilab/>

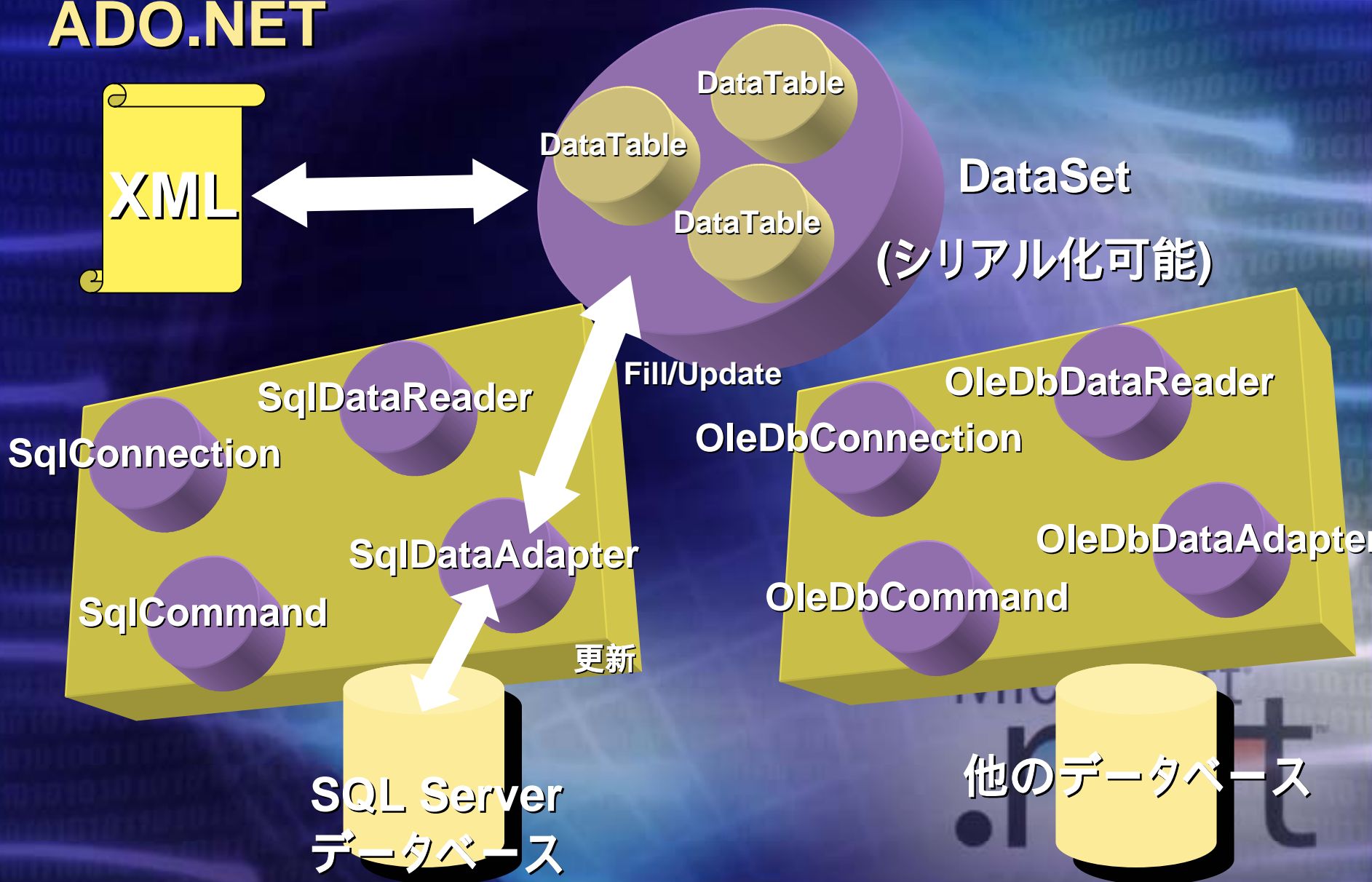
- Web Services Interoperability Org.

- Web サービスの相互運用性を促進するイニシアティブ
- ガイダンスと検証用ツールの提供
- <http://www.ws-i.org/>

Microsoft
.net™

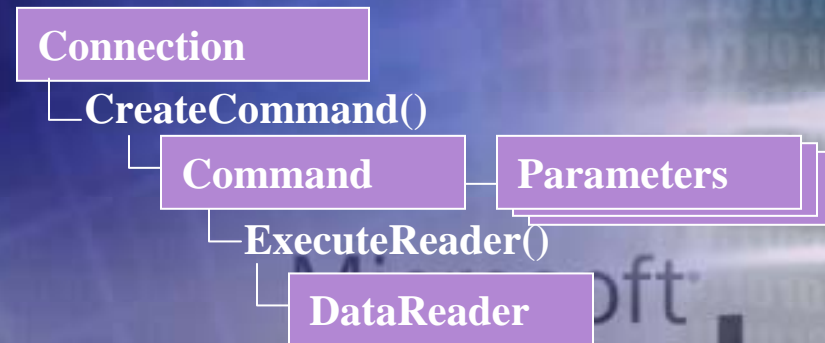
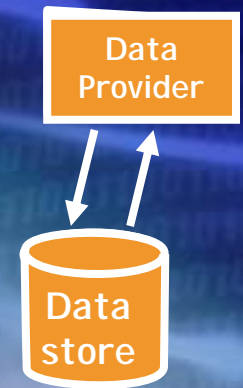
.NETのデータアクセス技術

ADO.NET



マネージプロバイダ

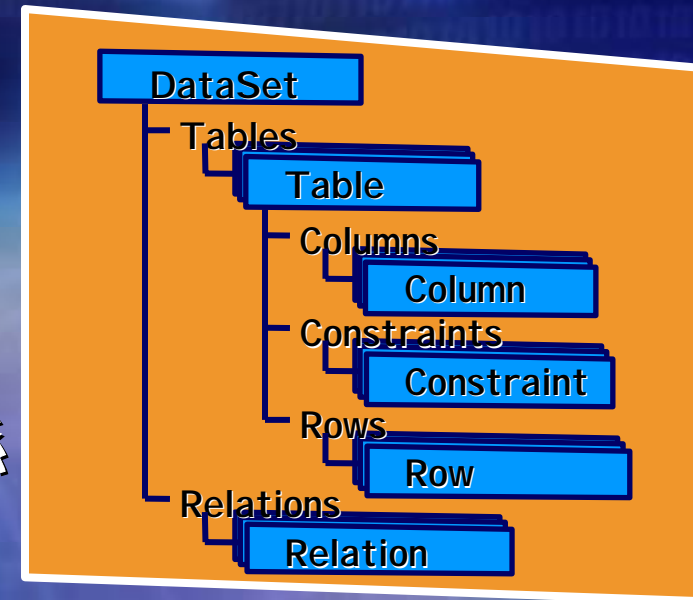
- データソースとのインタラクションを管理
 - OLE DB レイヤと同等 (マネージ環境)
 - 直接コンシューマ インターフェイスを公開
 - COM/オートメーションの二分法を解消
 - データストアの全インターフェイスに対応するものではない
 - 特定のデータソースに最適化
- 3つの標準プロバイダを提供
 - System.Data.OleDb
 - System.Data.Odbc
 - System.Data.SqlClient



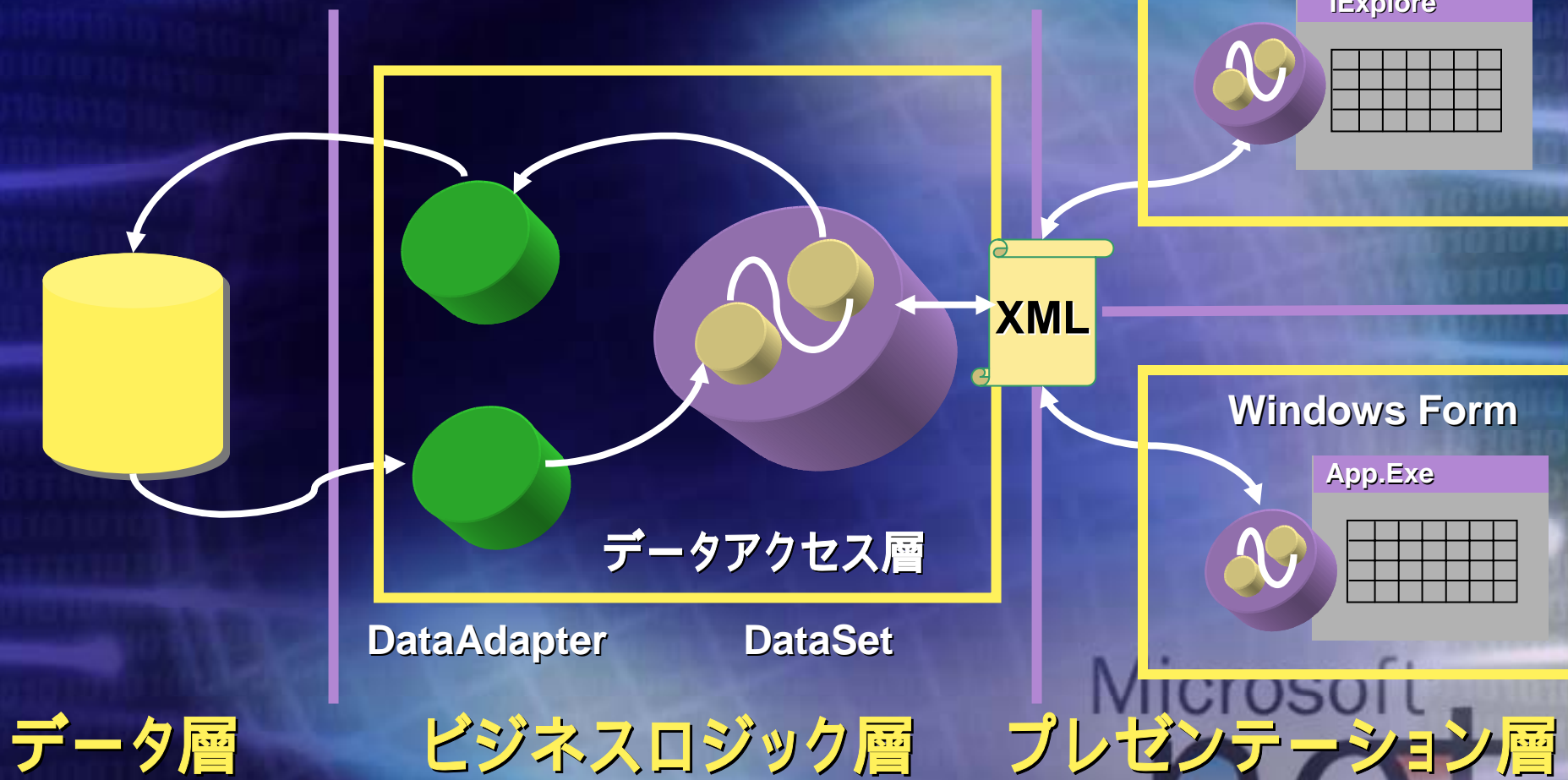
DataSet

ユニバーサルデータコンテナ

- データのリレーショナルなビュー
 - テーブル、カラム、行、制約、関係
- 直接メタデータの作成とデータの挿入が可能
- 明示的な非接続モデル
 - 非接続、リモート可能なオブジェクト
 - データソースやプロパティに関する知識を持たない
 - 共通のふるまい
 - 予測可能なパフォーマンス性能
 - 配列ライクなインデクス付け
 - 強い型付け



データへのアクセス 非接続モデル



Microsoft
.net

型つきDataSet

- 型のない場合 (ADO も同様):

```
Dim rowCustomer As System.Data.DataRow

For Each rowCustomer In MyDataSet.Tables("Customers").Rows
    Console.WriteLine(rowCustomer.Fields("CompanyName"))
Next
```

- 型がある場合:

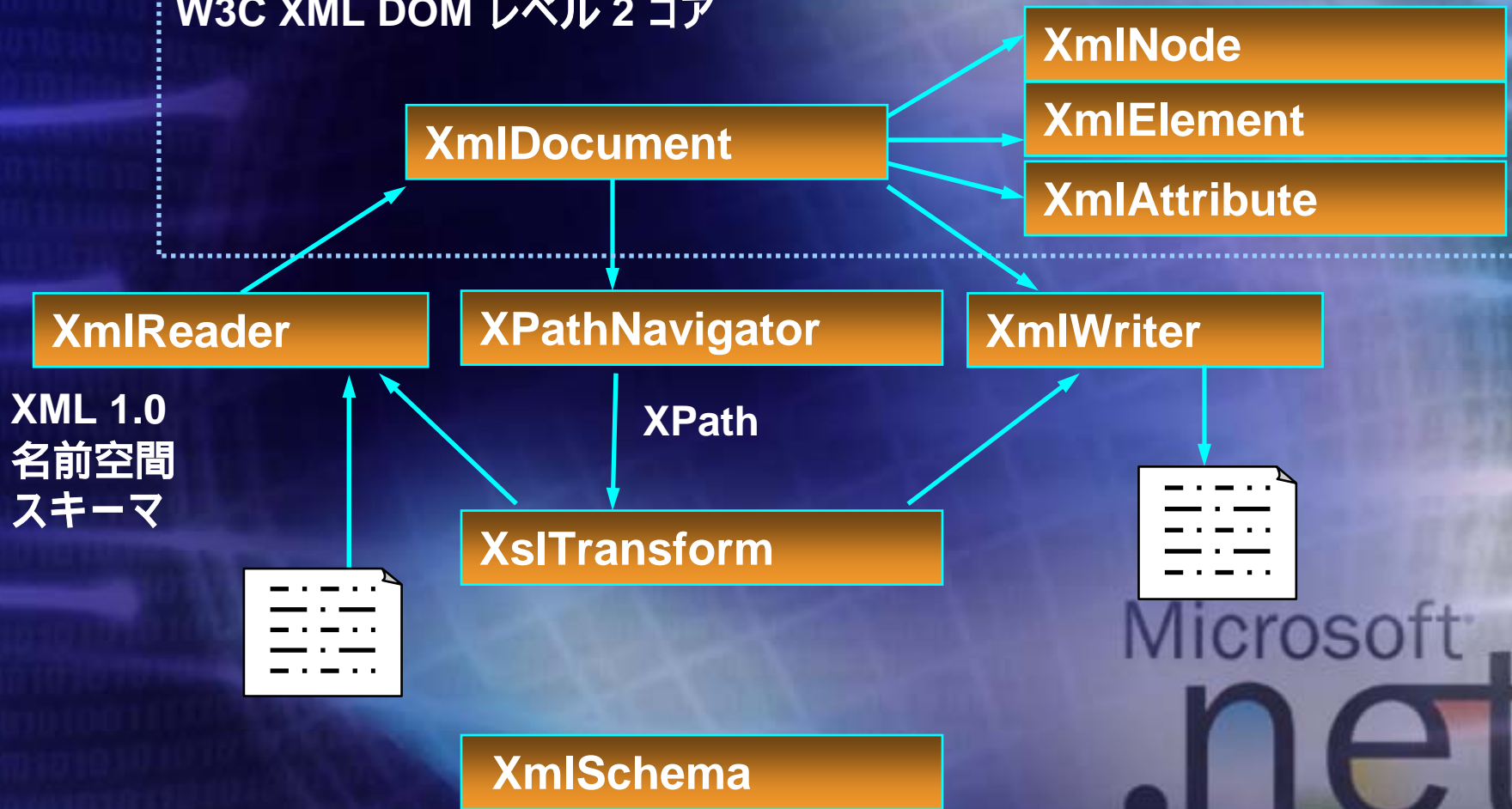
```
Dim rowCustomer As System.Data.DataRow

For Each rowCustomer In MyDataSet.Customers.Rows
    Console.WriteLine(rowCustomer.CompanyName)
Next
```

.net

XMLフレームワーク アーキテクチャ

W3C XML DOM レベル 2 コア



Microsoft
.net

SQLXML 概要

- リレーショナルデータのXML ビューの提供
- フラットなリレーショナルデータの半構造化、階層的なビュー
- 2つのビュー:クエリと更新
- 複数のアクセス方法
(HTTP、ADO、ADO.NET、SOAP)
- ミドル層とサーバーサイド
- XMLベース: 拡張可能、プラットフォーム非依存のデータフォーマット

クエリ/更新

- **FOR XML (raw, auto, nested, explicit)**
 - 行セットをXML形式で取得するSQL言語拡張
- **XML ビュー – XPath, XQuery**
 - XMLファイルのようにリレーショナルデータを処理(注釈つきスキーマを利用)
- **Updategrams/Diffgrams**
 - XMLビューによりデータ更新
- **Bulkload**
 - 大容量のXMLファイルを既存テーブルへ分割・格納

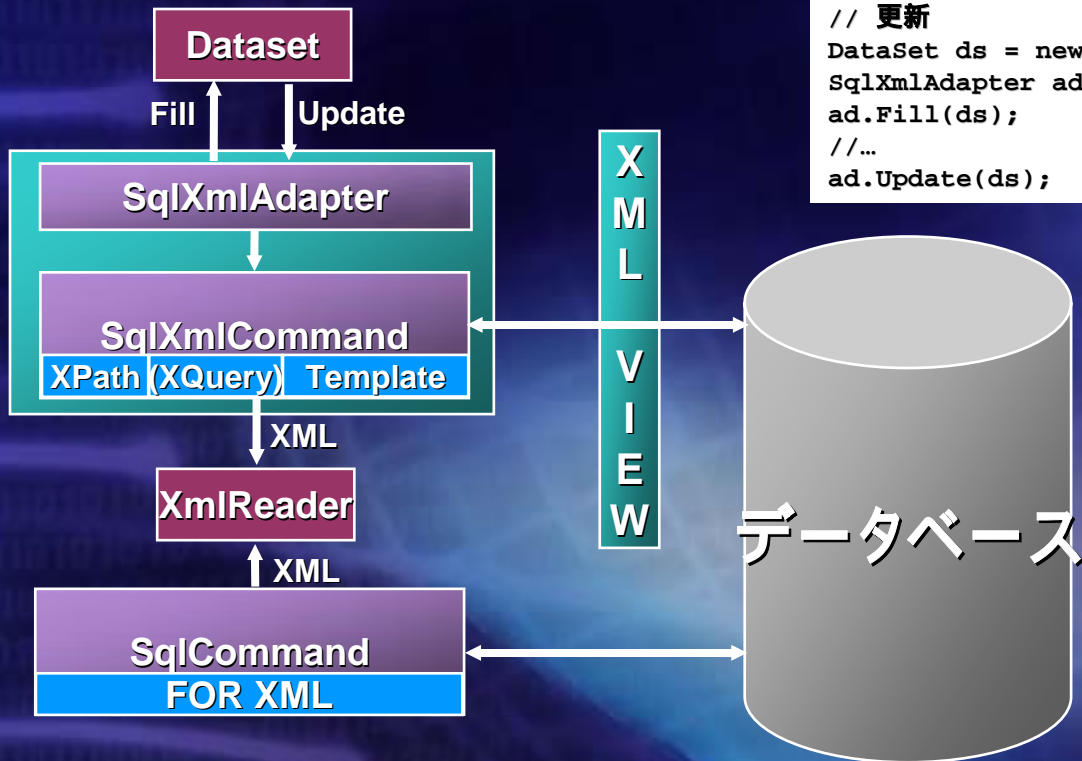
アプリケーションシナリオ例

データ駆動によるXML Web サービス

- C/Sベースアプリケーションを最低限の手間でXML Webサービス化する
 - ビジネスロジックとデータはSQL Serverに格納のまま
 - 既存のデータアクセスAPIとスキーマはほとんど変更なし
- 拡張可能な、疎結合アーキテクチャ
 - XML マッピング層へのクエリとXSLT変換によりWebページを生成
 - クライアントはSOAPでもアクセス可能

.NET と SQL Server 連携

- SQLXML
マネージプロバイダ
 - Microsoft.Data.SqlXml
- SQLXML 3.0 (Microsoft SQL Server 2000 Web Services Toolkit)に収録



```
// 読み取り
SqlXmlCommand cmd = new SqlXmlCommand(ConnStr);

cmd.CommandType = SqlXmlCommandType.XPath
cmd.CommandText = "/Customer[@CustID='ALFKI']";
cmd.SchemaPath = "nwind.xml";
cmd.RootTag = "ROOT";
cmd.XslPath = "Customers.xsl";

XmlReader r = cmd.ExecuteXmlReader();

// 更新
DataSet ds = new DataSet();
SqlXmlAdapter ad = new SqlXmlAdapter(cmd);
ad.Fill(ds);
//...
ad.Update(ds);
```

Microsoft
.net

型つき DataSet の進化

ADO.NET

DataReader

DataSet

ResultSet

Rel

Xml

Obj

Microsoft
.net

ObjectSpaces

- データのオブジェクトビュー
 - 永続オブジェクトとしてデータを公開
 - EJB、JDOのような
 - データソースに依存しない
 - SQL Server、OLE DBストア、XML ファイルなど
 - 既存のデータアクセス (System.Data)、XML API (System.Xml) の上に構築

ObjectSpaces によるコード例

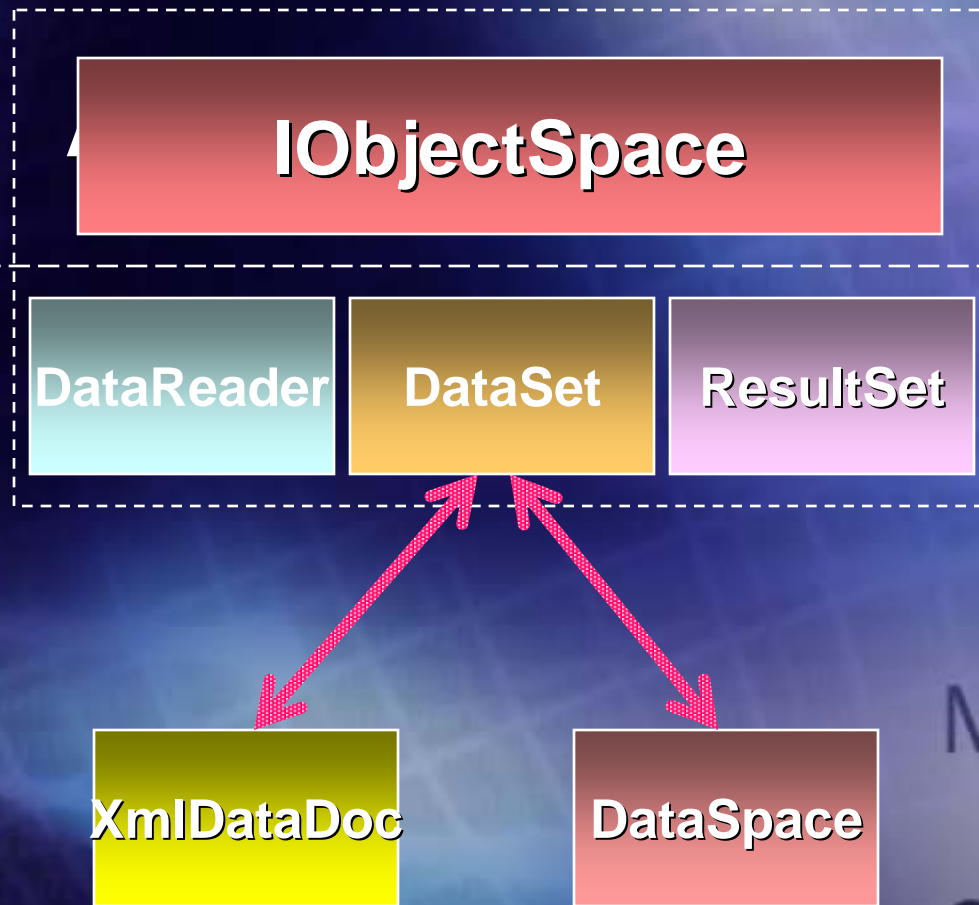
```
string q = "SELECT * FROM CUSTOMERS WHERE STATE = 'WA'";  
SqlConnection c =  
    new SqlConnection("myserverConnection");
```

```
CustomerDs ds = new CustomerDs();  
SqlDataAdapter ad = new SqlDataAdapter(q,c);  
ad.Fill(ds,"Customer");
```

```
DataSpace space = new DataSpace(ds);  
float sum=0;  
foreach(Customer c in space.GetObjects(typeof(Customer), "City = Seattle" ))  
{  
    sum += c.Account.Amount;  
}
```

Microsoft
.net

ObjectSpaces 暗黙的モデル



暗黙的モデル
の追加

Microsoft
.net

ObjectSpaces 暗黙的コードの例

```
IObjectSpace space = ObjectSpaceFactory.CreateObjectSpace(“source.xml”);
```

```
float sum=0;  
foreach(Customer c in space.GetObjects(typeof(Customer), “City = Seattle” ))  
{  
    sum += c.Account.Amount;  
}
```

Microsoft
.net

マッピングの例

```
<map>
  <type name="Customer" dataSource="cus" source="Example">
    <property name="Id" dataSource="CustomerId" />
    <property name="Name" dataSource="ContactName" />
    <property name="Title" dataSource="ContactTitle" />
    <property name="State" dataSource="State" />
    <property name="Orders" relationship="CustomerOrders"
      side="Parent" />
  </type>
  <relationship name="CustomerOrders" type="OneToMany"
    parentType="Customer" childType="Order">
    <key parent="CustomerId" child="CustomerId" />
  </relationship>
</map>
```

Microsoft
.net

アーキテクチャ

ObjectSpaces の暗黙的
モデル

ObjectSpace

Xml ObjectSpaces の明示的
モデル

モデル

DataSpace

DataSet

Managed Providers

TEL

単一のエントリポイント

- IObjectSpace

- **CreateObject** – オブジェクトの新規作成
- **DeleteObject** – オブジェクトの削除
- **Update/UpdateAll** – オブジェクトの更新
- **Begin/Commit/RollbackTransaction** – トランザクションの管理
- **GetObject/GetObjects** – 条件に従ったオブジェクト(複数も可)の取得

OPath

オブジェクトのクエリ言語

- オブジェクトモデルの上に定義
 - 開発者にとってなじみやすい
 - 複雑なクエリも可能
- Customer テーブルへのクエリ例
 - `Orders.Details.Quantity > 50`
 - `Orders[ShippedDate > RequiredDate]`
 - `Orders[Freight > 1000].Details.Quantity > 30`

リレーションシップナビゲーション例

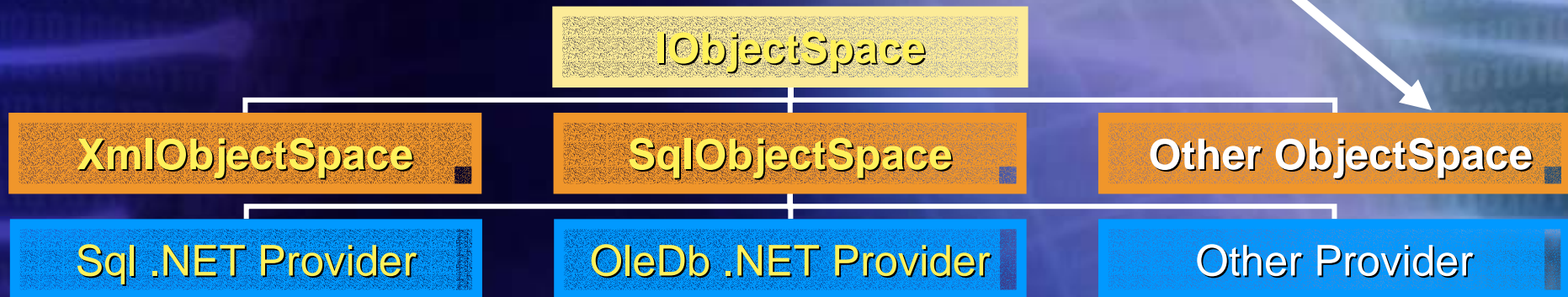
```
Customer c =  
    (Customer) os.GetObject(typeof(Customer), "Name='Mark Hill'");  
  
float sum = 0;  
  
foreach(Account a in c.Accounts)  
{  
    sum += a.Amount;  
}
```

オブジェクトの編集

```
Customer c =  
    (Customer) os.GetObject(typeof(Customer), "Name='Mark Hill'");  
  
float incomeTaxRate = 0.3;  
foreach(Account a in c.Accounts) {  
    a.Amount -= a.Amount * incomeTaxRate;  
}  
  
Account a = (Account) os.CreateObject(typeof(Account));  
a.Amount = 100;  
c.Accounts.Add(a);  
  
os.UpdateAll();
```

プラグイン可能なアーキテクチャ

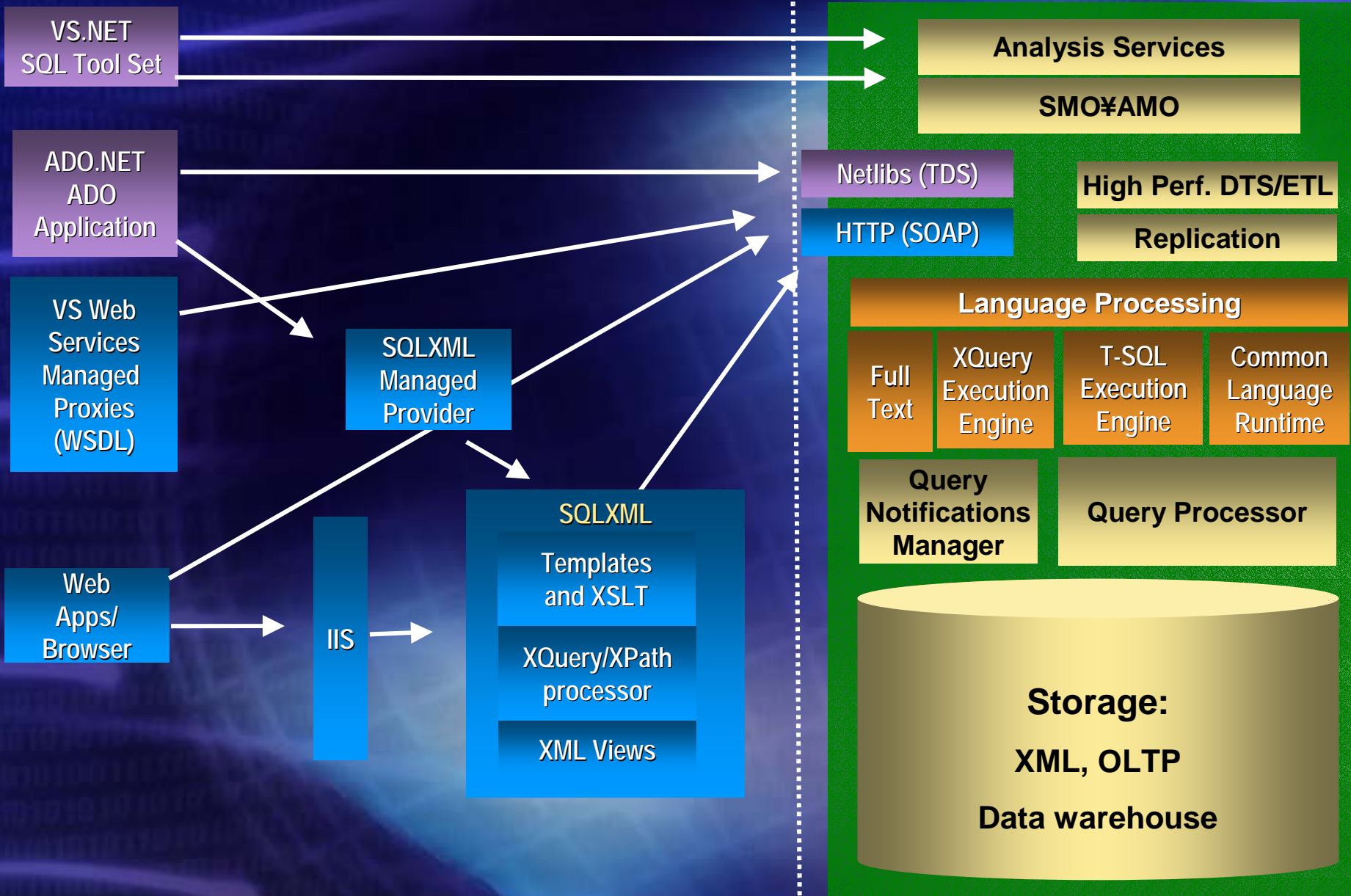
カスタム ObjectSpace



.NET データプロバイダ

.net

SQL Server "Yukon"



SQL Server / CLR 統合

基本インフラストラクチャ

- SQL Server の CLR ホスティング
 - 4Ss: Safety, security, scalability, speed
 - プロセス内におけるタイプセーフコードの実行
 - .NET framework 言語の活用 (VB, C#, C++)
- プロセス内でのデータアクセス
 - ADO .NET ベース
 - ミドル層と同じプログラミングモデル
- SQLTypes
 - マネージコード内での SQL タイプセマンティック

まとめ

.NET プラットフォーム概説

- インターネットによるビジネスの進化
 - 第3世代 Web ソリューションの時代へ
- XML Web サービスによる新たなソリューション
 - Web アプリケーションにおける革新
 - システム / アプリケーション統合の新形態としての利用
 - インターネット標準技術採用による相互運用/接続性の実現
- .NET プラットフォームを支える .NET Framework
 - アプリケーションフレームワークとして
 - 実行環境と基本クラス、高機能なクラスライブラリの提供
 - Visual Studio .NET による開発生産性の向上

MSDN アカデミック アライアンス

- マイクロソフト製品の最新リリース(ベータ版も含む)をご提供
- 最新およびバージョンアップ製品の自動送付、任意数インストール可
- 4年制大学の情報工学等「学科」「研究室」の教員・学生対象、教育・研究目的
- 1年間のサブスクリプション プログラム
 - 2002年4月1日～2003年3月31日
 - 年会費:118,000円/年
- 4月1日提供開始
- 登録お申し込みなど詳細情報
<http://www.msdnaa.com/japan/>

Microsoft
.net™

參考資料

Microsoft
.net™

Web リソース

- **MSDN Online Japan**
 - <http://www.microsoft.com/japan/developer/>
- **MSDN .NET Information**
 - <http://www.microsoft.com/japan/developer/net/>
- **GotDotNet (米国 .NET コミュニティーサイト)**
 - <http://gotdotnet.com/>
- **@IT .NET フォーラム(ニュース・テクニカルコンテンツなど)**
 - <http://www.atmarkit.co.jp/fdotnet/>
- **@IT .NET リンク集**
 - <http://www.atmarkit.co.jp/fwin2k/rescenter/indexpage/net.html>
- **.NET 関連書籍情報**
 - <http://www.microsoft.com/japan/developer/net/books.asp>

Microsoft
.net™