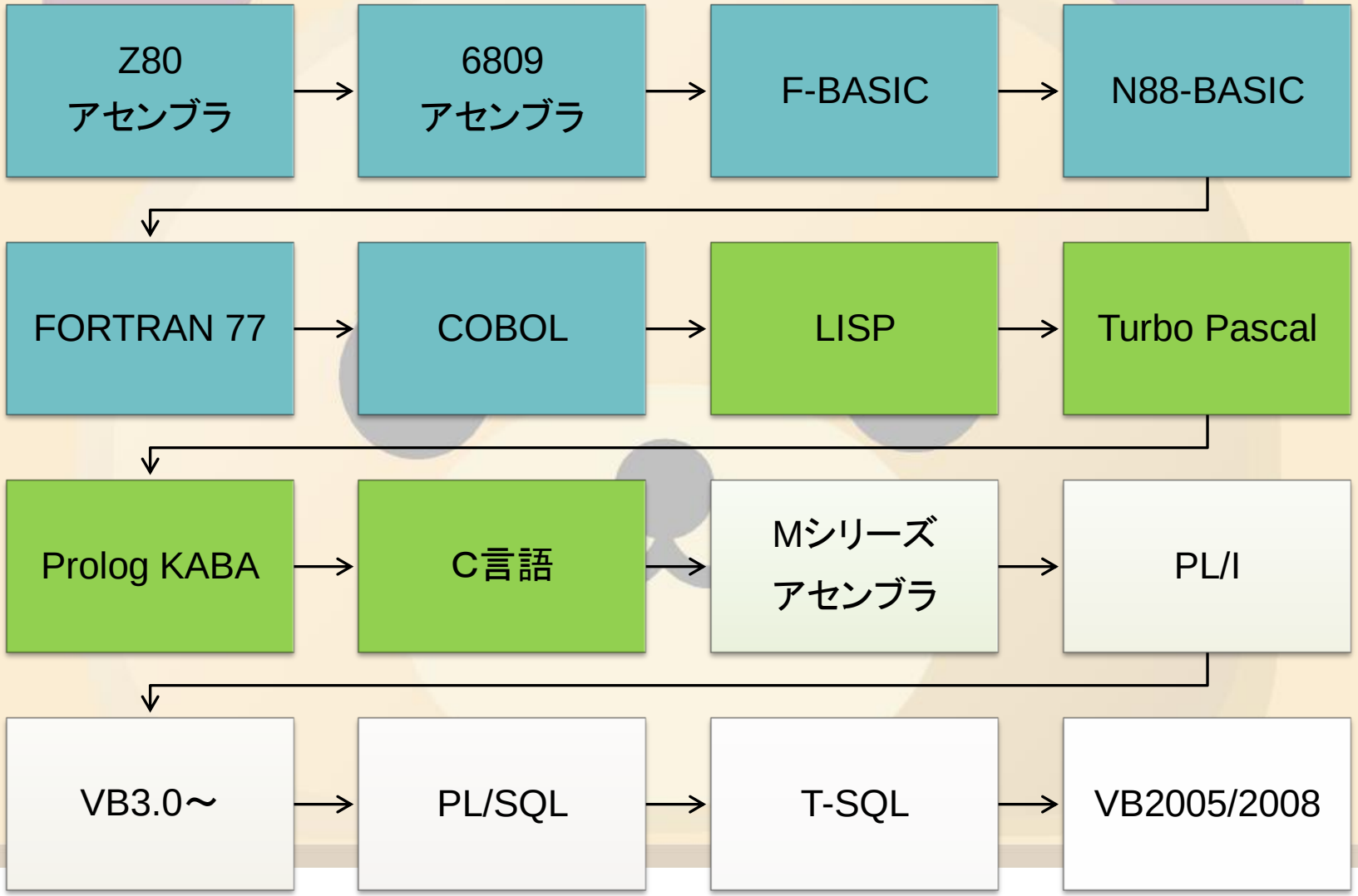
Miku + Oracle = Visual Basic ?

2008.06.07

初音 玲

自己紹介


Z80
アセンブラ
6809
アセンブラ
F-BASIC
N88-BASIC
FORTRAN 77
COBOL
LISP
Turbo Pascal
Prolog KABA
C言語
Mシリーズ
アセンブラ
PL/I
VB3.0～
PL/SQL
T-SQL
VB2005/2008

タイトルの意味は？

Part.2

Visual Basic 2008
+
Oracle Database 11g Release 1

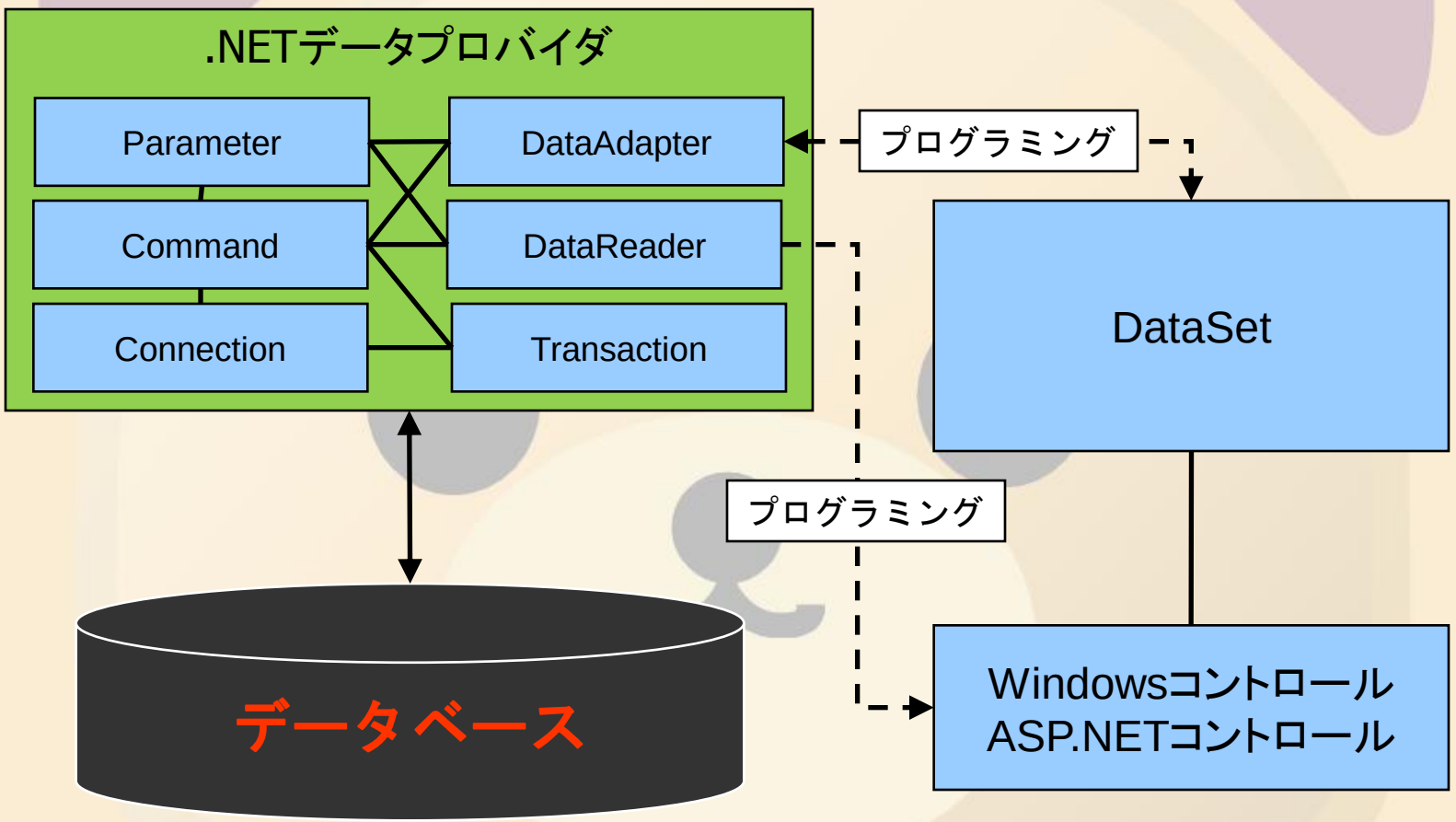
2008.06.07

初音　玲

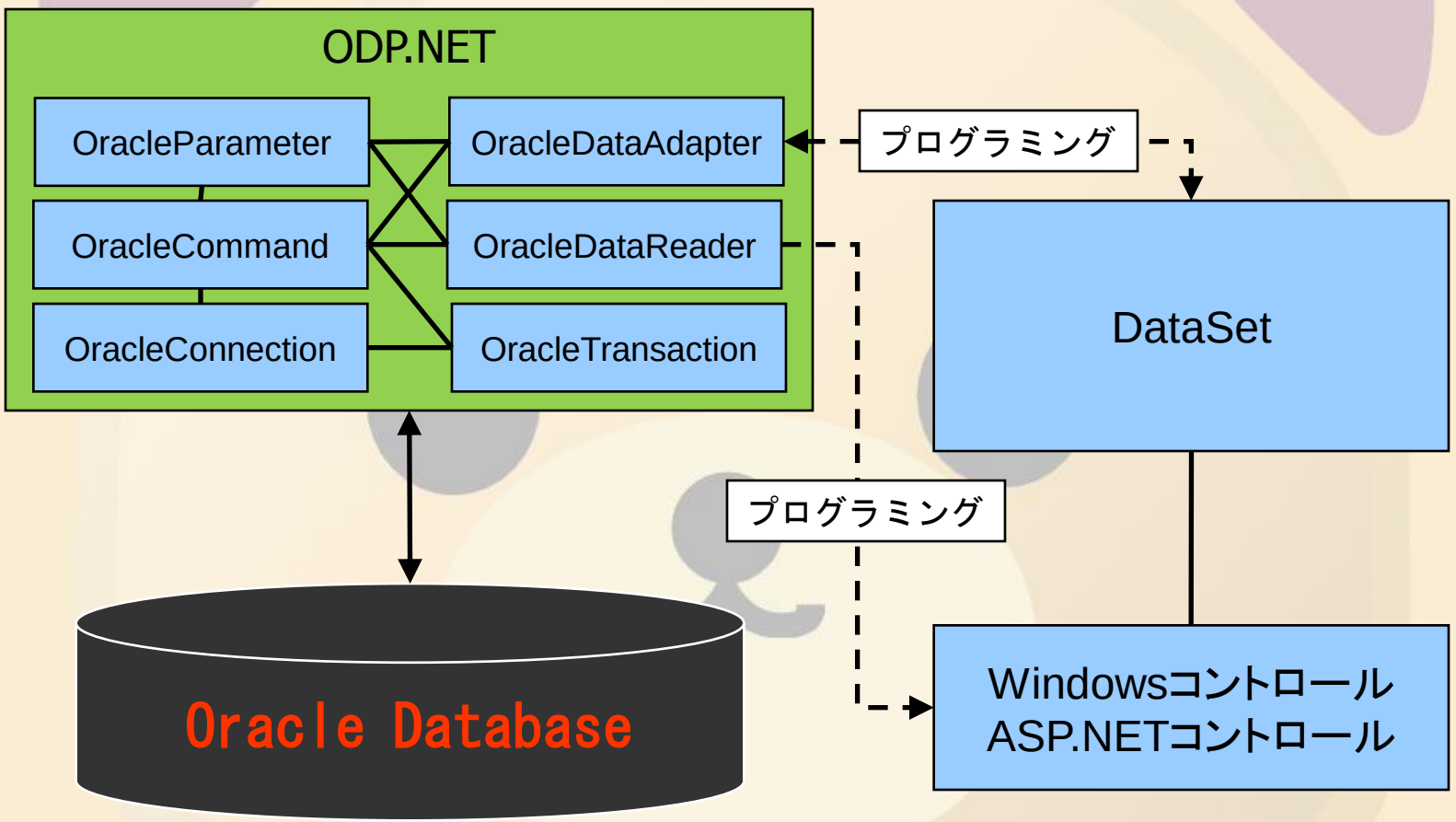
index

- 接続
- データ取得
- データ更新
- 権限

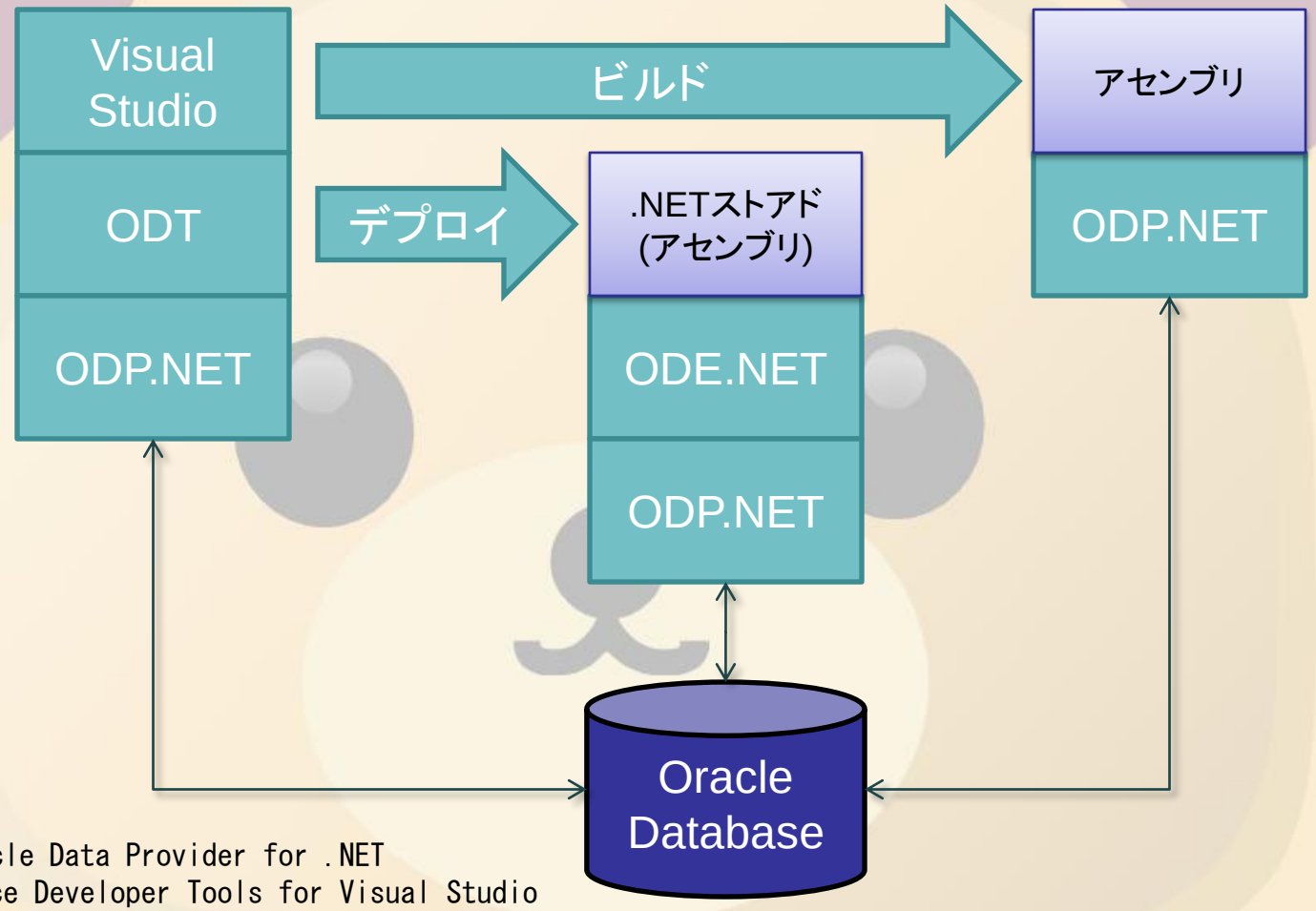
ADO.NETの基本的な構造



ODP.NETの基本的な構造



Visual StudioからOracleに接続



ODP.NET:Oracle Data Provider for .NET
ODT　　:Orace Developer Tools for Visual Studio
ODE.NET:Oracle Database Extensions for .NET

ODTとVSのバージョン関連表

	対応DBバージョン	対応Visual Studio
ODT 10.1.x.x	Oracle 9i Database Release 2 Oracle Database 10g Release 1	Visual Studio .NET 2003
ODT 10.2.0.1	Oracle 9i Database Release 2 Oracle Database 10g Release 1 Oracle Database 10g Release 2	Visual Studio .NET 2003
ODT 10.2.0.2	Oracle8i R8.1.7.4以降	Visual Studio .NET 2003 Visual Studio 2005
ODT 11.1.0.6	Oracle 9i Database Release 2～	Visual Studio .NET 2003 Visual Studio 2005 Visual Studio 2008

～ODT 10.2:サーバーエクスプローラとは別
ODT 11.1～:サーバーエクスプローラに統合

専用ミドルウェアは、やっぱり速い

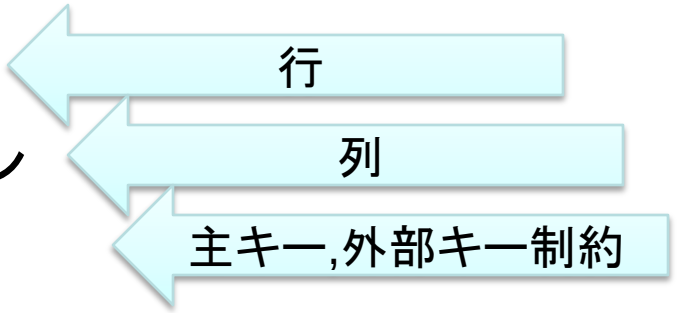
	OLE DB .NET Data Provider	OLE DB Provider for Oracle	ODP.NET	oo4o
ログオン	2.4	3.0	3.2	1.0
参照	2.0	3.5	3.5	1.0
追加	0.7	1.0	1.0	1.0
更新	0.7	1.2	1.2	1.0

VB2005+oo4oの処理時間を1としたときの相対比

独自プログラムによる測定

DataSetクラス

- メモリ上の仮想データベース
- DataTablesコレクション
  - DataTableクラス
    - DataRowsコレクション
    - DataColumnsコレクション
    - Constraintsコレクション
- DataRelationsコレクション
  - RDBMSのリレーション定義に相当
  - 親子関係を定義



ODP.NET

OracleConnection オブジェクト	• 特定のデータソースへの接続を確立
OracleCommand オブジェクト	• データソースに対してコマンドを実行
OracleDataReader オブジェクト	• データソースから前方向、読取専用でデータ取得
OracleDataAdapter オブジェクト	• DataSetを設定し、データソースを使用して更新内容を解決

index

Connectionオブジェクト

SQL Server　　SqlConnection

- Dim cn As New SqlConnection()
- cn.ConnectionString = "User Id=sa;" & _
  "Password=aU98rrx2;" & _
  "Initial Catalog=pubs;" & _
  "Data Source=servername;"
- cn.Open()

Oracle(ODP.NET)　　OracleConnection

- Dim cn As New OracleConnection()
- cn.ConnectionString = "User Id=scott;" & _
  "Password=tiger;" & _
  "Data Source=orcl.world;"
- cn.Open()

接続文字列の設定タイミング

変数宣言時点

- Dim cn As New OracleConnection("User id=・・・・")
- cn.Open

任意

- Dim cn As New OracleConnection()
  :
  (中略)
  :
- cn.ConnectionString = "User id=・・・・"
- cn.Open()

ADO.NETからのエラーの取得

Try～Catch

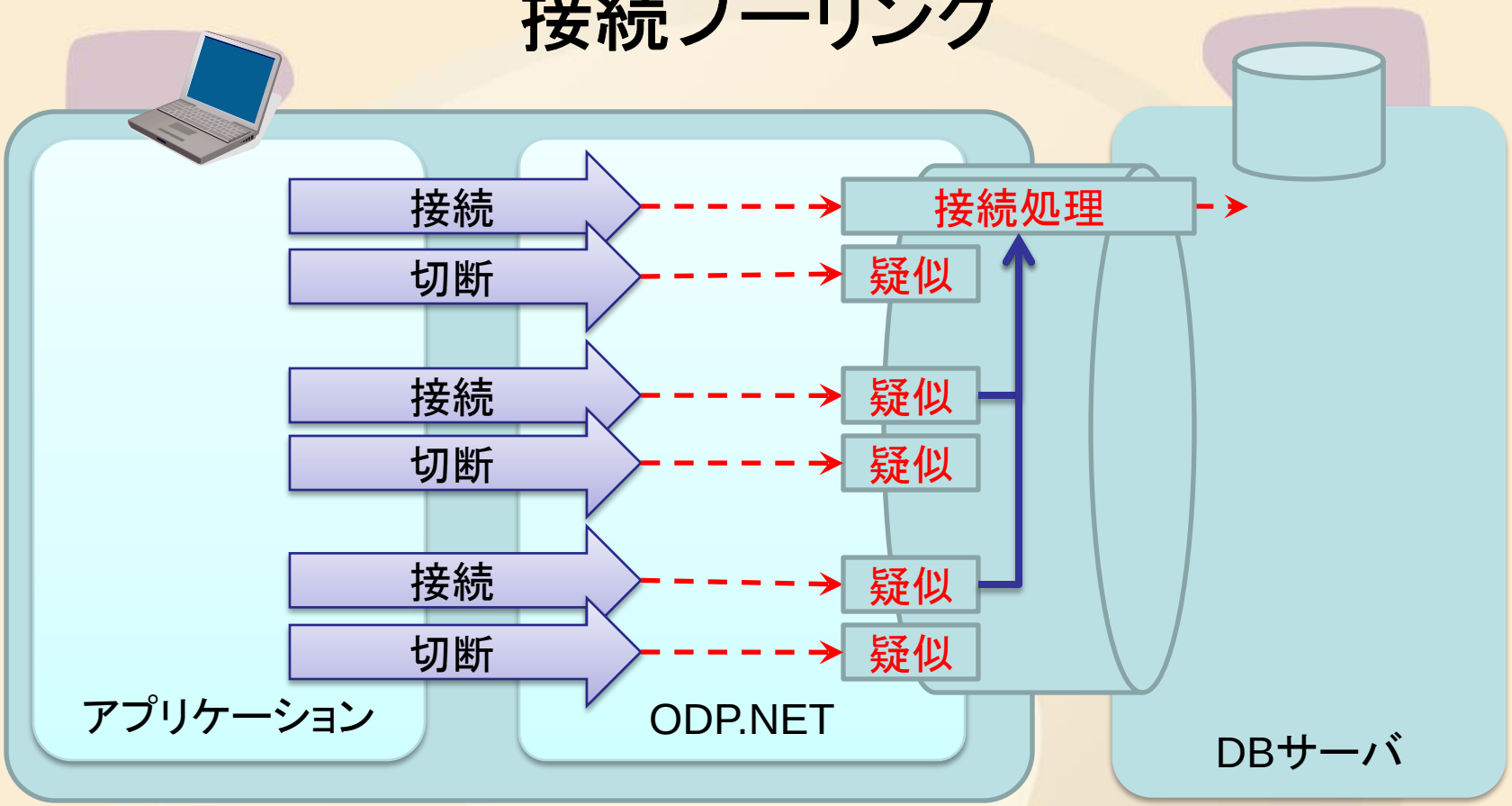
- cn.OpenをTry～Catchで囲む
- CatchにはOracleException？

OracleException

- ex.Message
- ex.StackTrace

Exception
で
いいかも？

接続プーリング



User Id={0};Password={1};Data Source=ホスト文字列;

- ODP.NETのデフォルト動作はPooling=True

index

Commandオブジェクト

SQL Server　　SqlCommand

- Using _cmd As New SqlCommand()
- 　_cmd.Connection = cn
- 　_cmd.CommandText = "SELECT * FROM employee"
- 　　:
- End Using

Oracle(ODP.NET)　　OracleCommand

- Using _cmd As New OracleCommand()
- 　_cmd.Connection = cn
- 　_cmd.CommandText = "SELECT * FROM employee"
- 　　:
- End Using

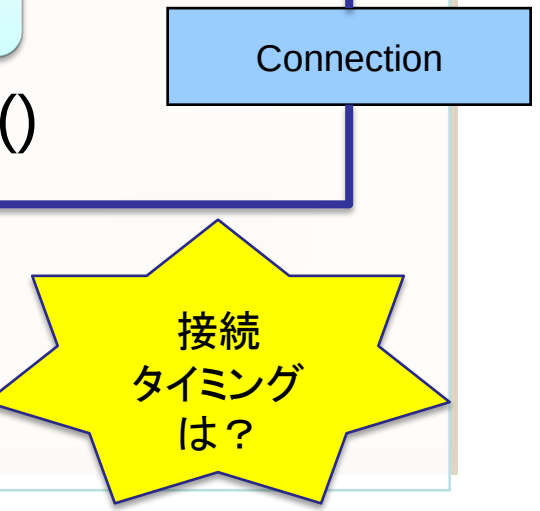
SELECT文設定タイミング

変数宣言時点

- Using _cmd As New OracleCommand(sql, cn)
  :
- End Using

任意

- Using _cmd As New OracleCommand()
- _cmd.Connection = cn
- _cmd.CommandText = sql
  :
- End Using



DataReaderオブジェクト

SQL Server　　SqlDataReader

- Dim rd As SqlDataReader()
- rd = _cmd.ExecuteReader
- Do While rd.Read
- 　Me.ResultList.Items.Add(rd.Item("fname").ToString)
- Loop
- rd.Close()

必須なのは？

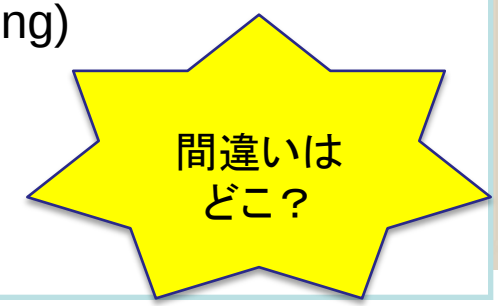
Oracle(ODP.NET)　　OracleDataReader

- Dim rd As OracleDataReader()
- rd = _cmd.ExecuteReader
- Do While rd.Read
- 　Me.ResultList.Items.Add(rd.Item("fname").ToString)
- Loop
- rd.Close()

必須なのは？

DataReaderを使う上での注意点

```
Using _cn As New OracleConnection(CnString)
    _cn.Open()
    Using _cmd As New OracleCommand(SqlString, _cn)
        Dim rd As OracleDataReader = Nothing
        rd = _cmd.ExecuteReader
        Do While rd.Read
            Me.ResultList.Items.Add(rd.Item("fname").ToString)
        Loop
    End Using
    Using _cmd As New OracleCommand(SqlString, _cn)
        Dim rd As OracleDataReader = Nothing
        rd = _cmd.ExecuteReader
        Do While rd.Read
            Me. ResultList.Items.Add(rd.Item("fname").ToString)
        Loop
    End Using
    _cn.Close()
End Using
```



列単位でデータを実取得する

```
rd = _cmd.ExecuteReader()
```

- rd.Readメソッドで1行分を読み込む
- 1行前の読みなおしは不可
- 行を飛ばして読み込むのも不可

```
rd = _cmd.ExecuteReader( _
CommandBehavior.SequentialAccess)
```

- rd.Item(0)で1項目分を読み込む
  rd.GetBytes(0,stp,outSize,0,bufferSize)で分割読み込み
- 先頭項目から順番に取得
- 項目取得後に、それより前の項目取得不可
- 項目を飛ばして取得不可

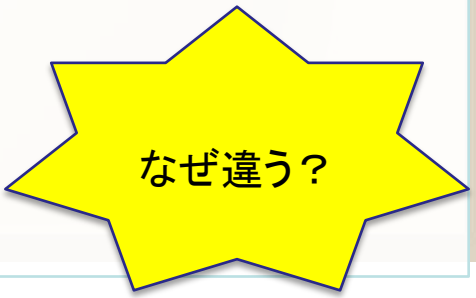
Parameterオブジェクト

SQL Server　　SqlParameter

```
sqlString = _"SELECT * FROM employee " & _
        "WHERE fname=@fname AND lname=@lname"
Using _cmd As New SqlCommand(sqlString, _cn)
    _cmd.Parameters.Add(New SqlParameter("fname", Me.FName.Text))
    _cmd.Parameters.Add(New SqlParameter("lname", Me.LName.Text))
                :
End Using
```

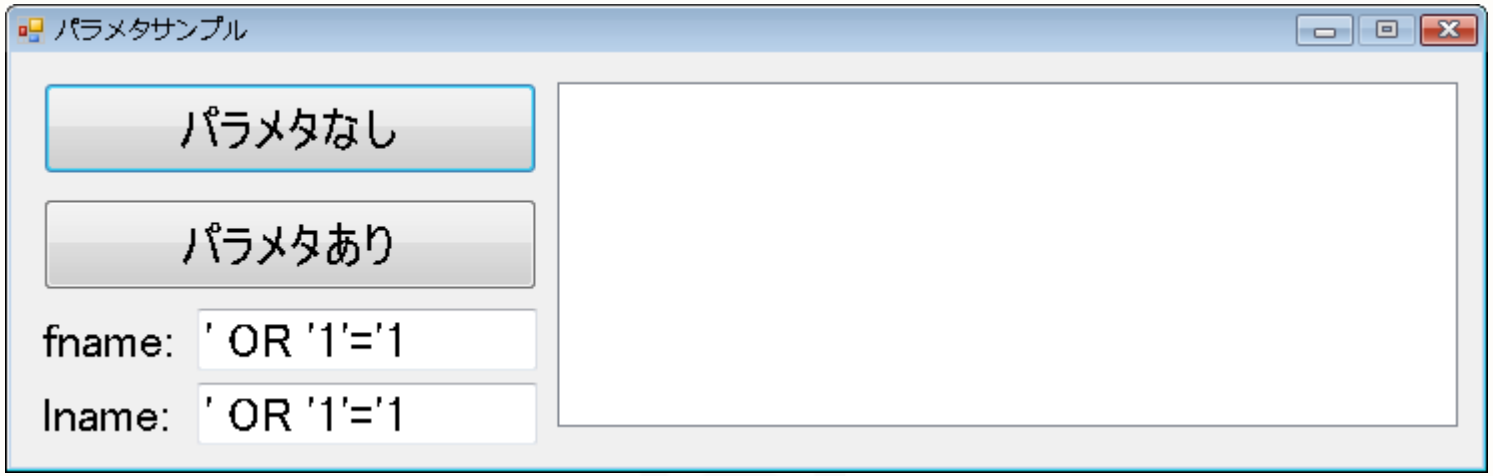
Oracle(ODP.NET)　　OracleParameter

```
sqlString = "SELECT * FROM employee " & _
        "WHERE fname=:fname AND lname=:lname"
Using _cmd As New OracleCommand(sqlString, _cn)
    _cmd.BindByName = True
    _cmd.Parameters.Add("fname", Me.FName.Text)
    _cmd.Parameters.Add("lname", Me.LName.Text)
                :
End Using
```



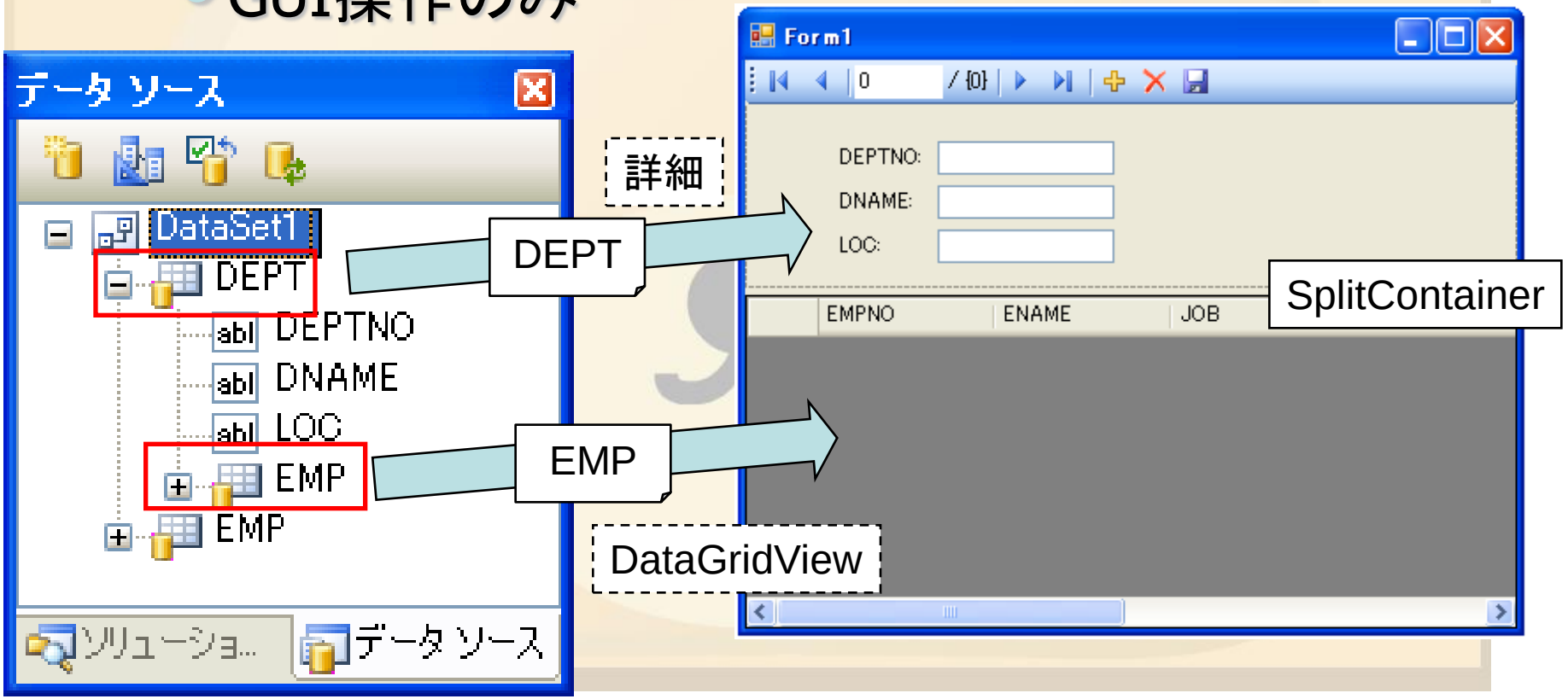
Parameterオブジェクト

以下の条件を与えたときの実行結果は？



マスターディテール形式の画面を作成

- GUI操作のみ



GUI操作による開発と生産性

GUIでお手軽開発で良いか検討

- すべてをカバーするものではない

意図したタイミングで意図したデータを取得

- 業務の特性に合わせて
- 画面の特性に合わせて
- お客様の特性に合わせて

そのためには

- やはりコーディングが必要

index

- 接続
- データ取得
- データ更新
- 権限

SQL文の直接実行(Commandオブジェクト)

SQL Server　　SqlParameter

```
sqlString = _
        "UPDATE employee SET minit=@minit WHERE emp_id=@emp_id"
Using _cmd As New SqlCommand(sqlString, _cn)
    _cmd.Parameters.Add(New SqlParameter("emp_id", Me.Emp_Id.Text))
    _cmd.Parameters.Add(New SqlParameter("minit", Me.Minit.Text))
    _cmd.ExecuteNonQuery
End Using
```

Oracle(ODP.NET)　　OracleParameter

```
sqlString = _
        "UPDATE employee SET minit=:minit WHERE emp_id=:emp_id"
Using _cmd As New OracleCommand(sqlString, _cn)
    _cmd.BindByName = True
    _cmd.Parameters.Add("emp_id", Me.Emp_Id.Text)
    _cmd.Parameters.Add("minit", Me.Minit.Text)
    _cmd.ExecuteNonQuery
End Using
```

データソースとDataSetクラスの対応付け

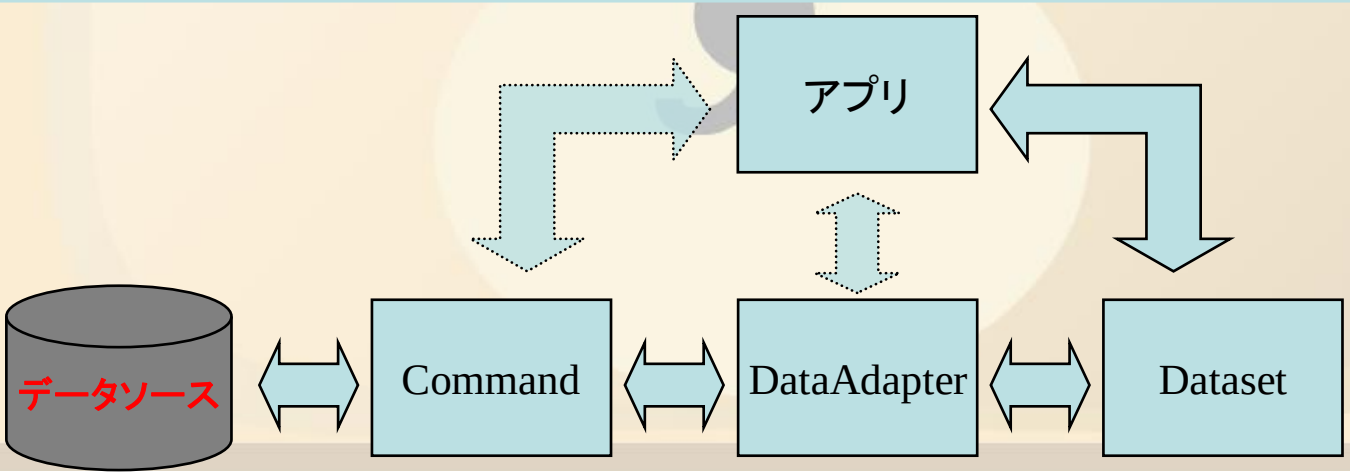
.NETデータプロバイダには、更新可能セットがない

DataSetは仮想的なデータベース

- もちろん更新も可能

.NETデータプロバイダとDataSetの相互乗り入れ

- DataSetによるデータソースの更新可能セットを実現



DataAdapterオブジェクト

SQL Server　　SqlDataAdapter

```
• Using _cn As New SqlConnection(CnString)
•     Using _cmd As New SqlCommand("SELECT * FROM employee", _cn)
•         Using _da As New SqlDataAdapter(_cmd)
•             _da.Fill(ds, "employee")
•             Me.ResultGrid.DataSource = ds.Tables("employee")
•         End Using
•     End Using
• End Using
```

Oracle(ODP.NET)　　OracleDataAdapter

```
• Using _cn As New OracleConnection(CnString)
•     Using _cmd As New OracleCommand("SELECT * FROM employee", _cn)
•         Using _da As New OracleDataAdapter(_cmd)
•             _da.Fill(ds, "employee")
•             Me.ResultGrid.DataSource = ds.Tables("employee")
•         End Using
•     End Using
• End Using
```

CommandBuilderでSQL作成

```
_cn.Open()
Using _tr As SqlTransaction = _cn.BeginTransaction()
    Using _cmd As New SqlCommand("SELECT * FROM employee ", _cn)
        _cmd.Transaction = _tr      '###重要###
        Using _da As New SqlDataAdapter(_cmd)
            Using _cb As New SqlCommandBuilder(_da)
                _da.UpdateCommand = _cb.GetUpdateCommand()
                _da.InsertCommand = _cb.GetInsertCommand()
                _da.DeleteCommand = _cb.GetDeleteCommand()
                Try
                    _da.Update(Ds, "employee")
                    _tr.Commit()
                Catch ex As Exception
                    MessageBox.Show(ex.Message, ･･･)
                    _tr.Rollback()
                End Try
            End Using
        End Using
    End Using
End Using
```

CommandBuilderオブジェクト

SetAllValuesプロパティ

- True:UPDATE SQLのSET句にすべてセット
- False:UPDATE SQLのSET句に変更値だけをセット
- 利用状況モニタで確認
  SET [minit] = @p1

実装すらされていません

ConflictOptionプロパティ

- CompareAllSearchableValues
  WHERE (([emp_id] = @p9) AND ([fname] = @p10) AND ((@p11 = 1 AND [minit] IS NULL) OR ([minit] = @p12)) AND ([lname] = @p13) AND ([job_id] = @p14) AND ((@p15 = 1 AND [job_lvl] IS NULL) OR ([job_lvl] = @p16)) AND ([pub_id] = @p17) AND ([hire_date] = @p18))
- CompareRowVersion
  WHERE (([emp_id] = @p9))
- OverwriteChanges
  WHERE (([emp_id] = @p9))

This property is not supported.

DataSetの利用時の注意点

SQL Serverの10進数型の扱い

- OracleのNUBER型の有効桁数
  - 38桁
- DataSet（CLR)の10進数型の有効桁数
  - 28桁

実行時エラー

- Fillメソッド実行時に10進数型の値が28桁を超えたとき

対策

- Oracle.DataAccess.Types.OracleDecimal

DataSetの利用時の注意点

Fillメソッドで全データ取得

selectメソッドで必要な情報取得

Fillメソッドで必要データのみ取得

表示データから選択データ抽出

selectメソッド利用

トランザクション処理

GUI操作で作成したDBアプリには致命的障害

- 保存ボタンで保存されるのは、DEPTのみ

```
Me.Validate()
Me.DEPTBindingSource.EndEdit()
Me.DEPTTableAdapter.Update(Me.DataSet1.DEPT)
```

EMPへのUpdateメソッドを追加で解決？

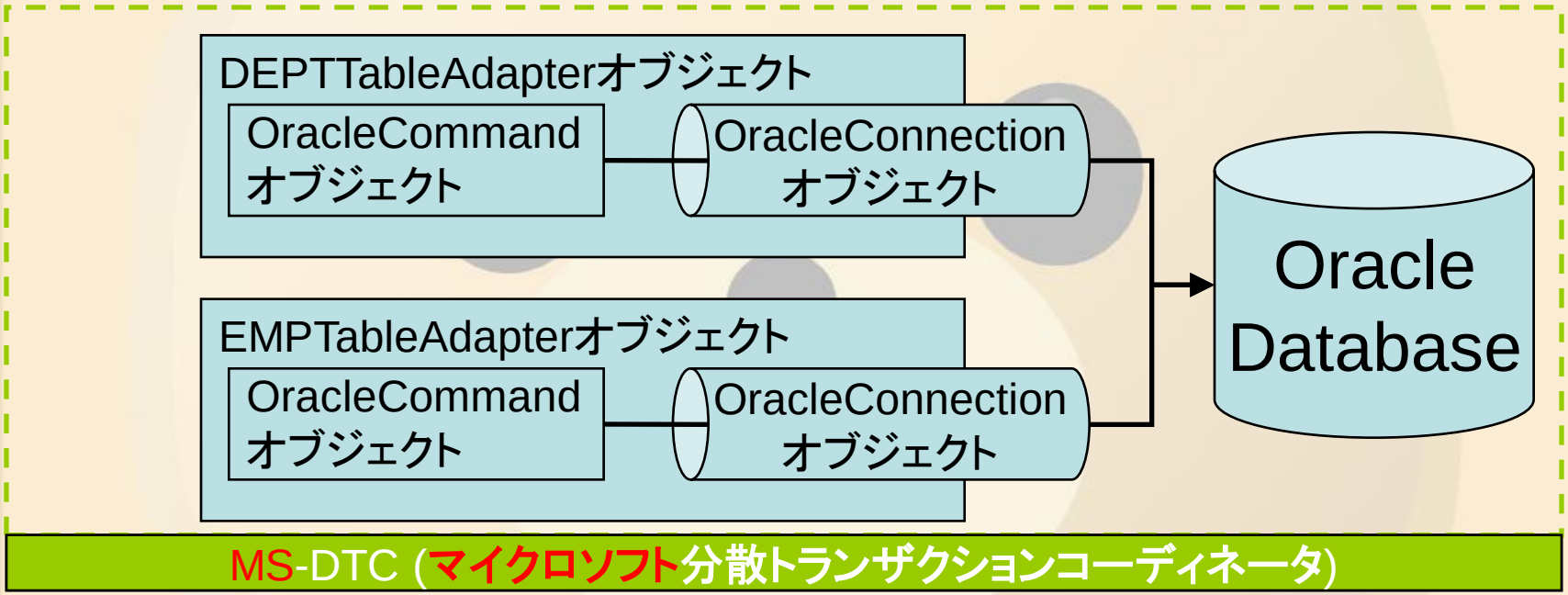
- DEPT更新後、EMP更新に失敗すると……

トランザクション制御が必要

暗黙的なトランザクション

TableAdapterごとにConnectionが存在

– Connectionに対するトランザクションでは無理



System.Transactionsクラスを使う

暗黙的なトランザクション

```
Try
    Me.Validate()
    Using _trn As New System.Transactions.TransactionScope
        Me.DEPTBindingSource.EndEdit()
        Me.DEPTTableAdapter.Update(Me.DataSet1.DEPT)
        '
        Me.EMPBindingSource.EndEdit()
        Me.EMPTableAdapter.Update(Me.DataSet1.EMP)
        '
        _trn.Complete() 'トランザクション完了
    End Using
Catch ex As System.Transactions.TransactionAbortedException
    MessageBox.Show(ex.Message)
Catch ex As Exception
    MessageBox.Show(ex.Message)
End Try
```

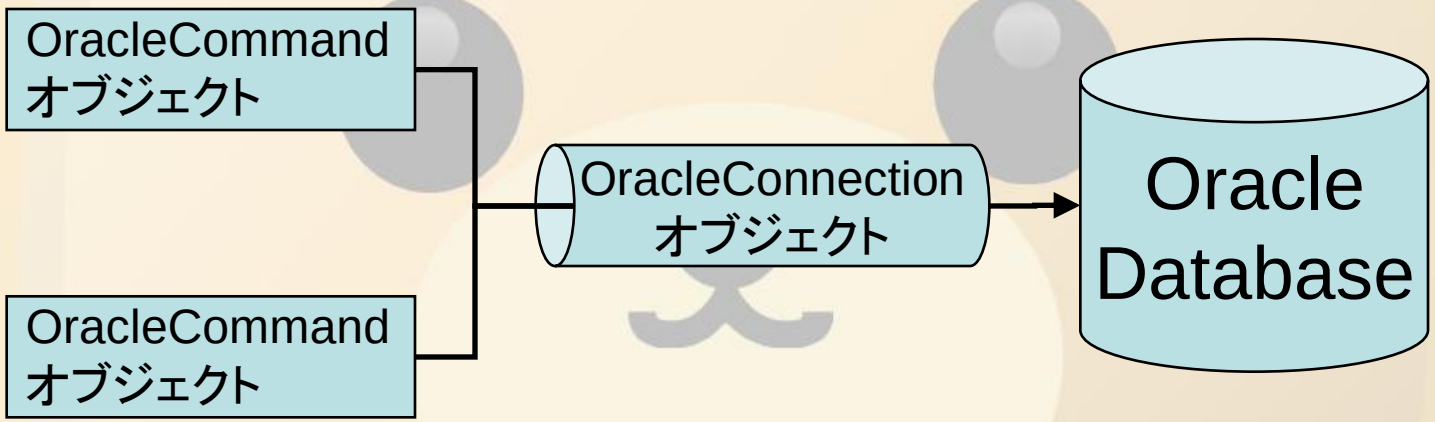
ADO.NET 2.0からは
COM+カタログ登録不要
GACへの登録不要

System.Transactionsの参照設定を忘れないように

明示的なトランザクション

Connectionが1つ

- Connectionに対するトランザクションで可能



明示的なトランザクション

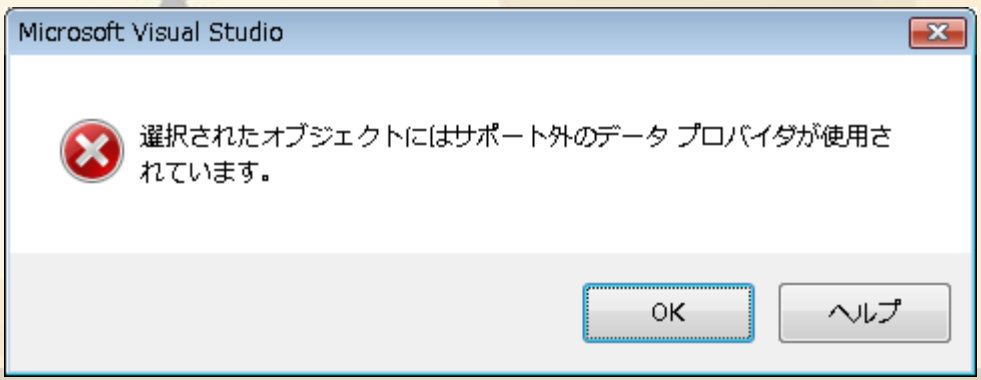
```
_trn = cn.BeginTransaction()
Try
                    :
    _DBCmdEMP.Transaction = _trn
    _DBCmdDEPT.Transaction = _trn
    _DBCmdEMP.ExecuteNonQuery()
    _DBCmdDEPT.ExecuteNonQuery()
    _trn.Commit()
Catch ex As Exception
    _trn.Rollback()
End Try
```

(おまけ)LINQ to SQL

LINQtoSQLクラスを追加(.dbml)

サーバエクスプローラからDrag&Drop

標準クエリ演算子を記述



index

権限

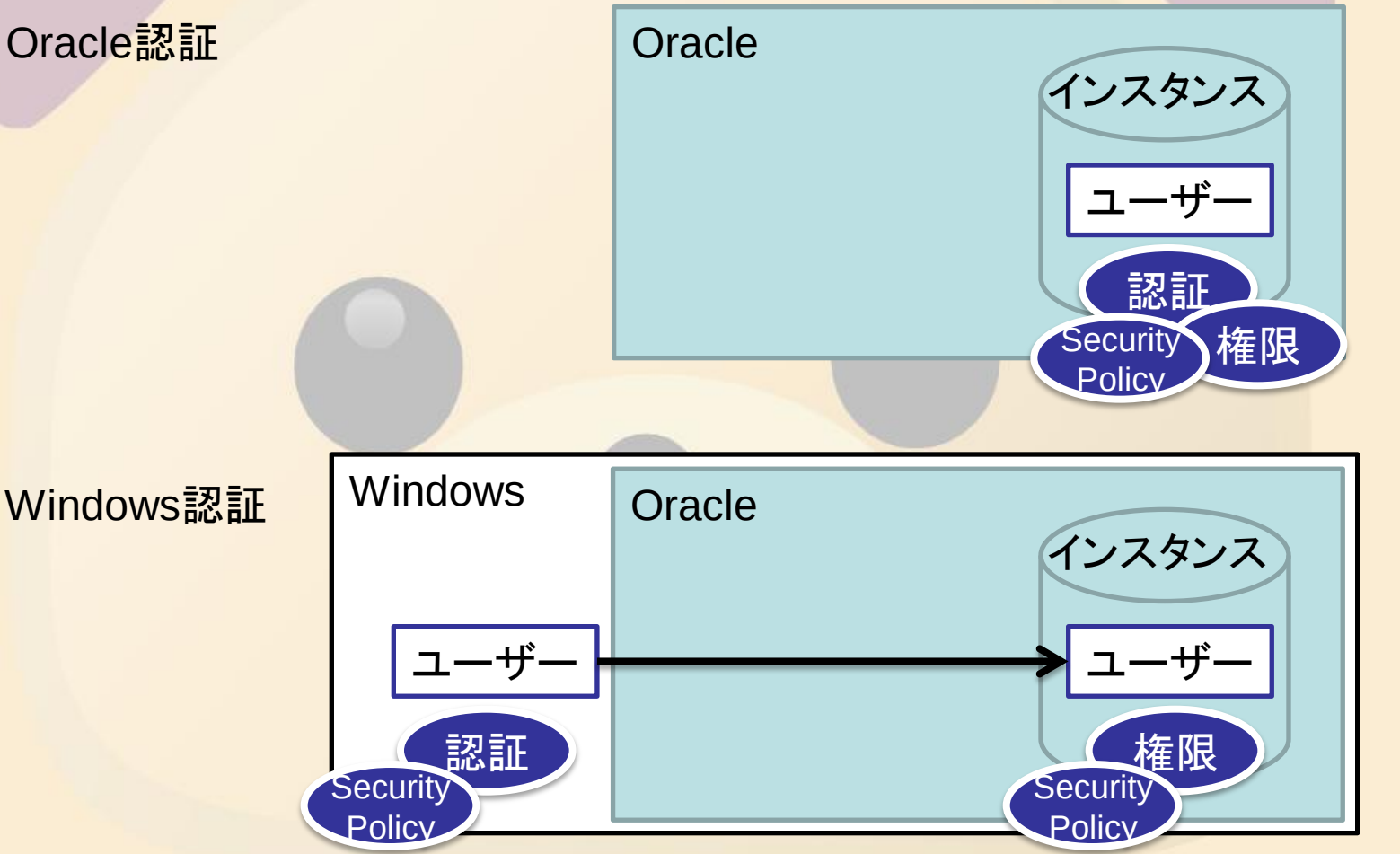
接続は、アプリ固有ユーザID

- コネクションプーリング
- アプリ側でアクセス制御
- 直接ツールで接続されたら？
- アプリ側にバグがあったら？

接続は、利用者固有ユーザID

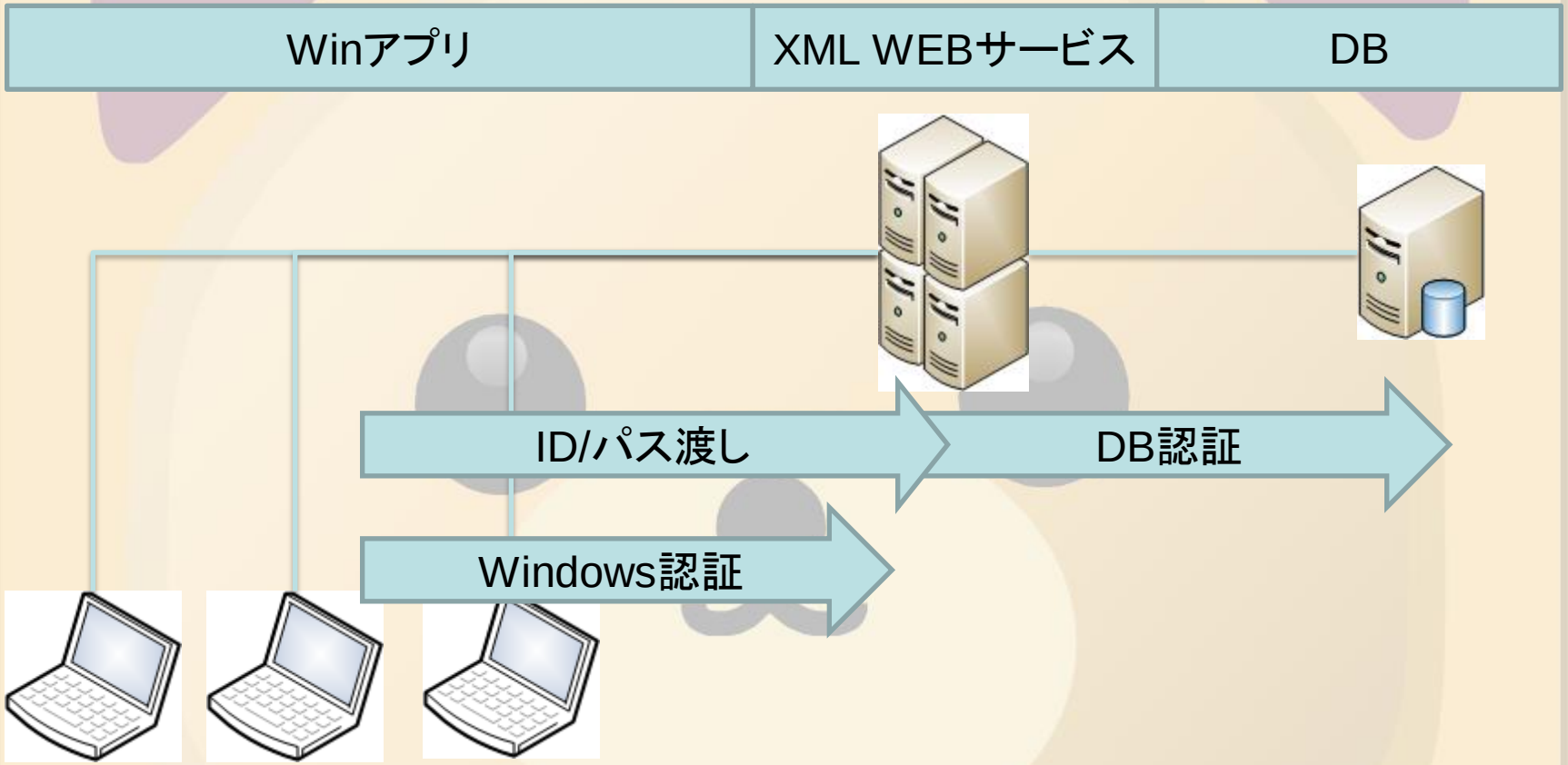
- コネクションプーリング
- DBの設定でアクセス制御
- 直接ツールで接続されても安全
- アプリ側にバグがあっても安全

Oracleにおけるユーザ管理

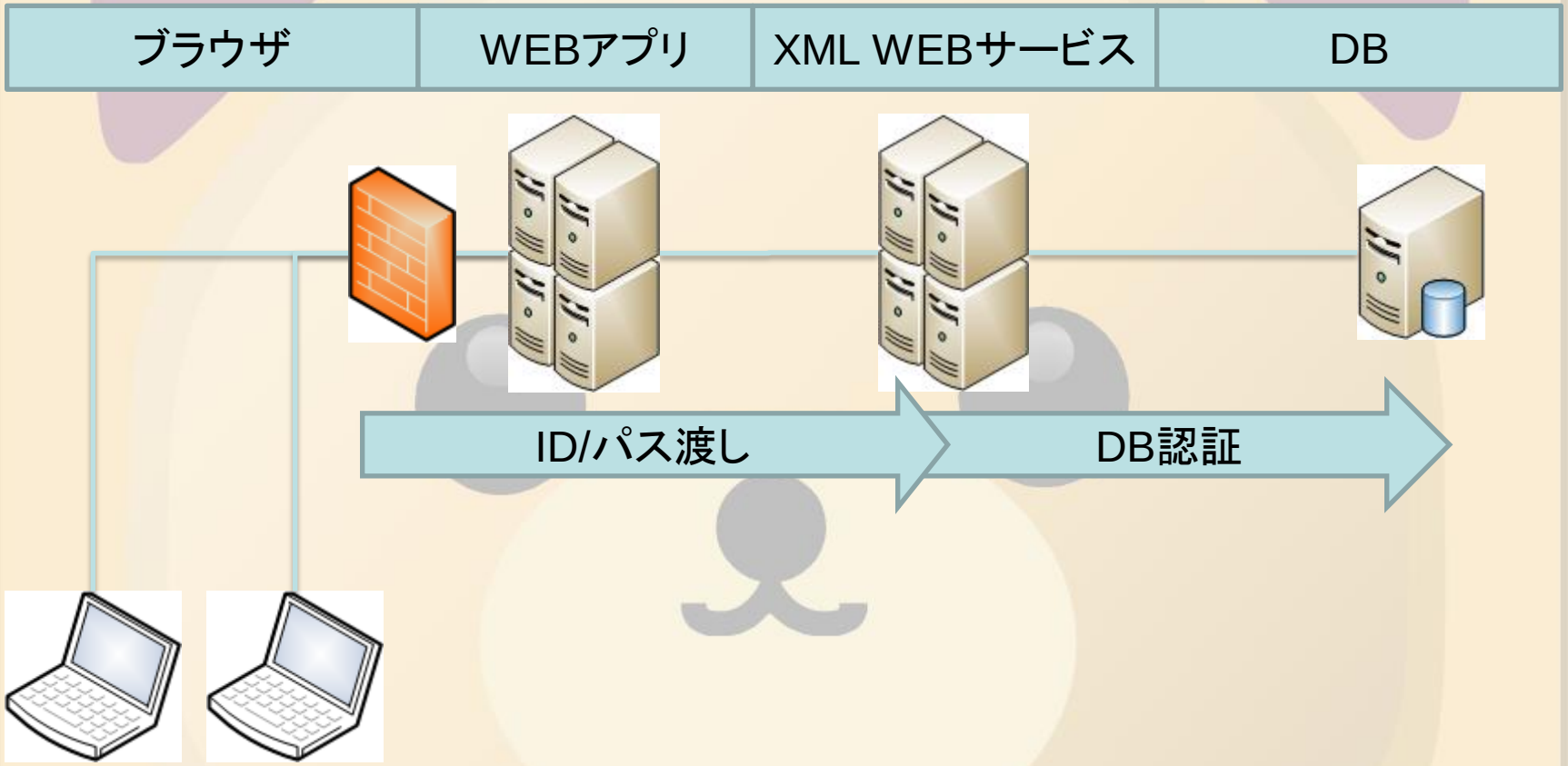


ローカル認証だと4万人くらいが限界なのでAD認証も考慮

Windowsアプリにおけるお勧め認証構造



WEBアプリにおけるお勧め認証構造



まとめ

- 接続
  - 接続プーリング
- データ取得
  - コード記述開発
  - GUI操作開発
- データ更新
  - CommandBuilder
  - トランザクション
- 権限
  - Oracle認証
  - Windows認証

QA