Introduction to OpenTouryo, an Application Framework for .NET

April 23, 2014

Opening

- OpenTouryo has been developed by Hitachi Solutions (Formerly Hitachi Systems & Services) in 2007.

- In the 7 years up to now, it has been adopted by many projects of this Company. (product development and contract development, and up to large-scale mission-critical systems from small departmental system)

- The system that need high quality and high reliability is need development infrastructure which is backed by a proven effective. The know-how that we have accumulated many years from NET application development has been aggregated to OpenTouryo.

Changes in IT technology is intense, and Choice of technology has increased in recent years. On the other hand, the importance of the use of IT is increasing more for the growth of the company. This time, Hitachi Solutions decided to publish as open source under the name OpenTouryo, It had been internal use as .NET application development framework.

Contents

Contents

1.1 Summary

OpenTouryo is an application framework of the full stack on the premise .NET Framework 2.0 or more.

Corresponding to various system, Allow the application development of high quality.
Web, C/S, batch, RIA, WebAPI, Embedded.etc

- document
  - Use guide, tutorial
  - English manual for offshore development
- license
  - source : Apache License, Version 2.0
  - document : Creative Commons - CC BY 2.1 JP

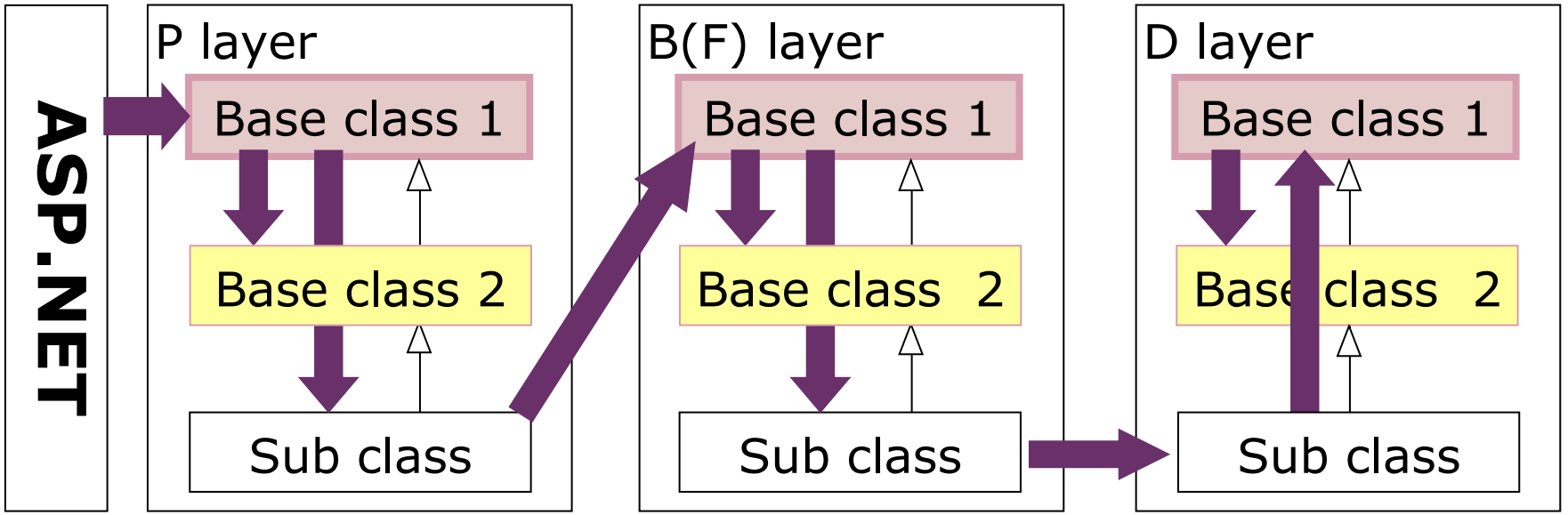
1.2 system requirements

		Product name
Development environment		• Microsoft Visual Studio 2010 – 2013 • Microsoft Visual C# 2010 – 2013 • Microsoft Visual Basic 2010 – 2013
Execution environment	**Run Time**	• .NET Framework 2.0 – 4.5.1 • ASP.NET2.0, 4.0(+ AJAX Extensions) • ASP.NET MVC 4 • Windows Azure SDK for .NET • Silverlight, Windows store application
	Data Provider	• .NET Framework Data Provider for SQL Server • OLEDB.NET Data Provider • ODBC.NET Data Provider • Oracle Data Provider for .NET • IBM DB2.NET Data Provider • HiRDB.NET data provider • MySQL Connector/NET • PostgreSQL Npgsql.NET data provider
	WWW Browser	• Internet Explorer Version 6.0, – 11.0 It is possible to use in other browsers as well if the dialog display function is not used (Support for mobile phone oriented CHTML).

Contents

2.1 Three-tier architecture that satisfy both of customization and versatility

It provides an architecture that rules the process flow of each layer,
By the base class of two-stage, juggle customization and versatility.



Provide common functionality to be used in base class2 (customization) and base class1 (fixed)

Authentication	Session management	Data access	Transaction management	Exception process
Permission	Message acquisition	Input check	Sub-screen display	Security

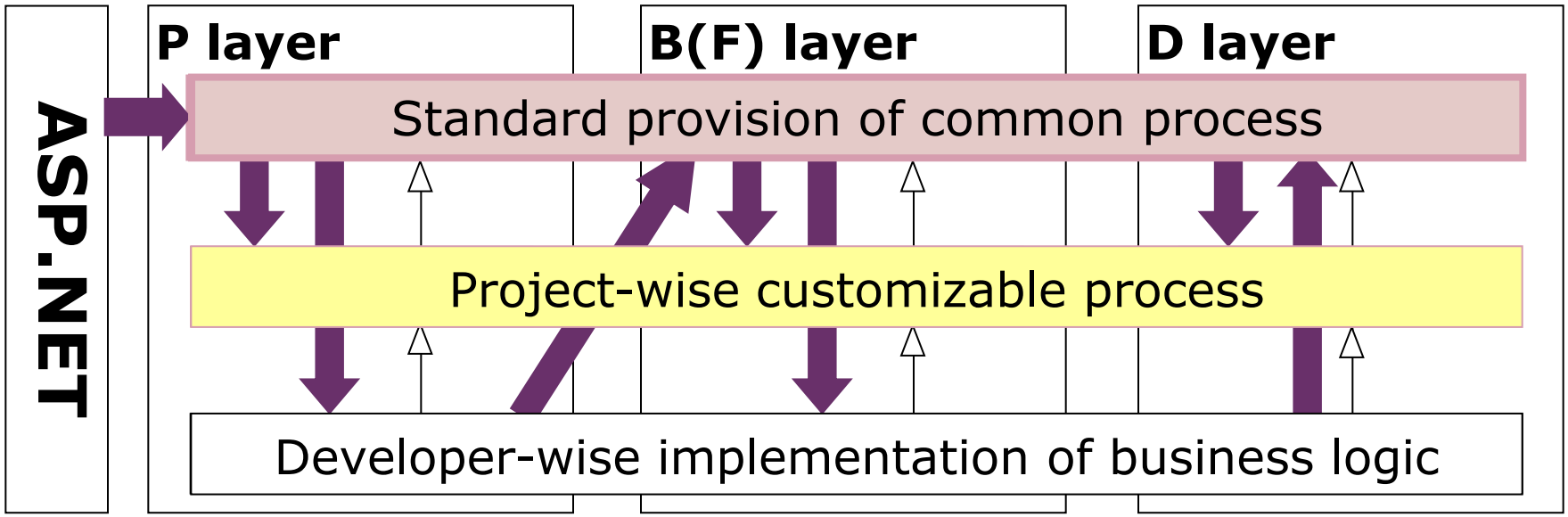
2.2 The function of each layer

Layer	Function
P	This structure dosen't spoil the operability of the designer in Visual Studio. ◆ Screen transition control, session management ◆ Deterrence of illegal operation, input check. Etc
B	◆ Flow Control To control the flow of basic processing(customizable) of the following ■ Start and end processing, exception handling ■ DB connection management, transaction management ■ log output, performance measurement .etc ◆Communication control function Provides remote processing mechanism between servers using various protocols. Compatible with a variety of system configurations, including cloud and Onpure by this.
D	◆ Data access libraries like MyBatis ■ Dynamic parameterized query ■ Dynamic parameterized query definition and verification tool ◆ Batch processing for SQL generation parts .etc

2.3 Components, tools

class	Function
Common libraries	◆ String editing, input check function ◆ Log output, share information and message management ◆ Internationalization ■ Local ⇔ UTC time conversion, ■ CultureInfo support of various messages resources ◆ Asynchronous event control, shared memory management
Custom Controls	◆ Corresponding to WindowsForms and WebForm ◆ String editing, input check function, Grid display in
Automatic generation tool	◆ D-layer automatic generation tool table CRUD ◆ Automatic generation tool of the table maintenance screen

2.4 Promotion of standardization by the OpenTouryo



Standardization of application architecture using “Touryo”

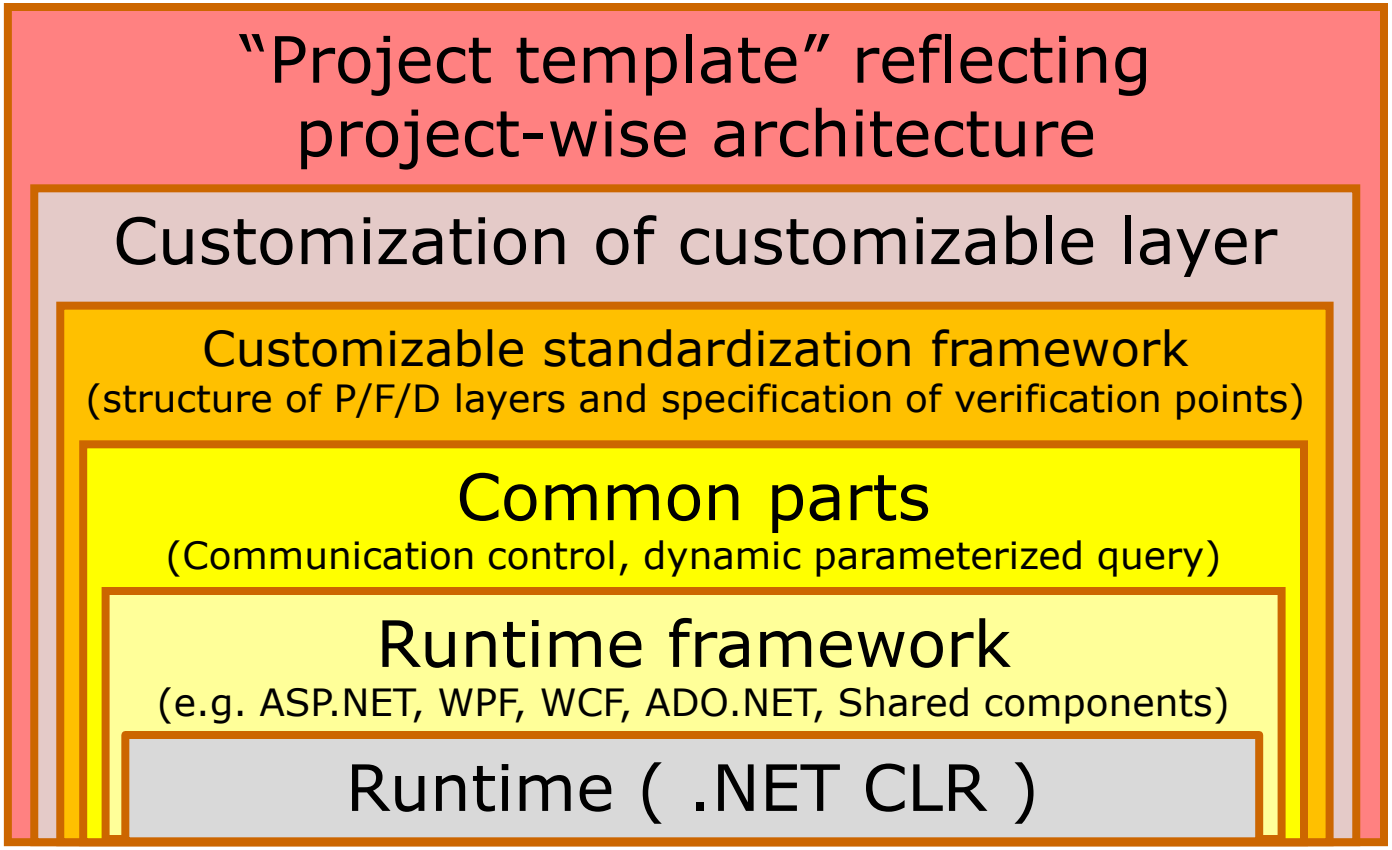
1. Standardization of all layers across the P/B/D layers, is possible.
2. Base process implementation is divided into base class 1 and 2.
   - Base class 1: Common process (Execution engine)
   - Base class 2: Customized Project-wise(Customer-wise)
3. Hence, developers can concentrate on implementing business logic into the sub class.

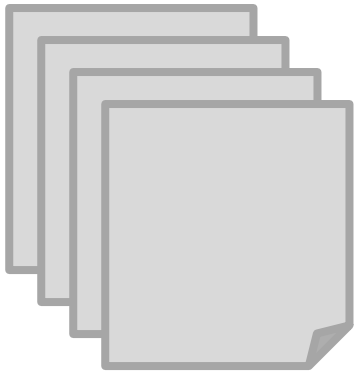
2.5 Project Template

Online processing and Batch processing that is customized to fit the architecture of the system of the project is Called “project template”.

Beforehand Prepares project template and deploy to the project, This allow a rapid start-up of development of projects.

template-base is published to facilitate the preparation of the project template. See the "Tutorial_Template_development.doc" for ways to use Template-base.



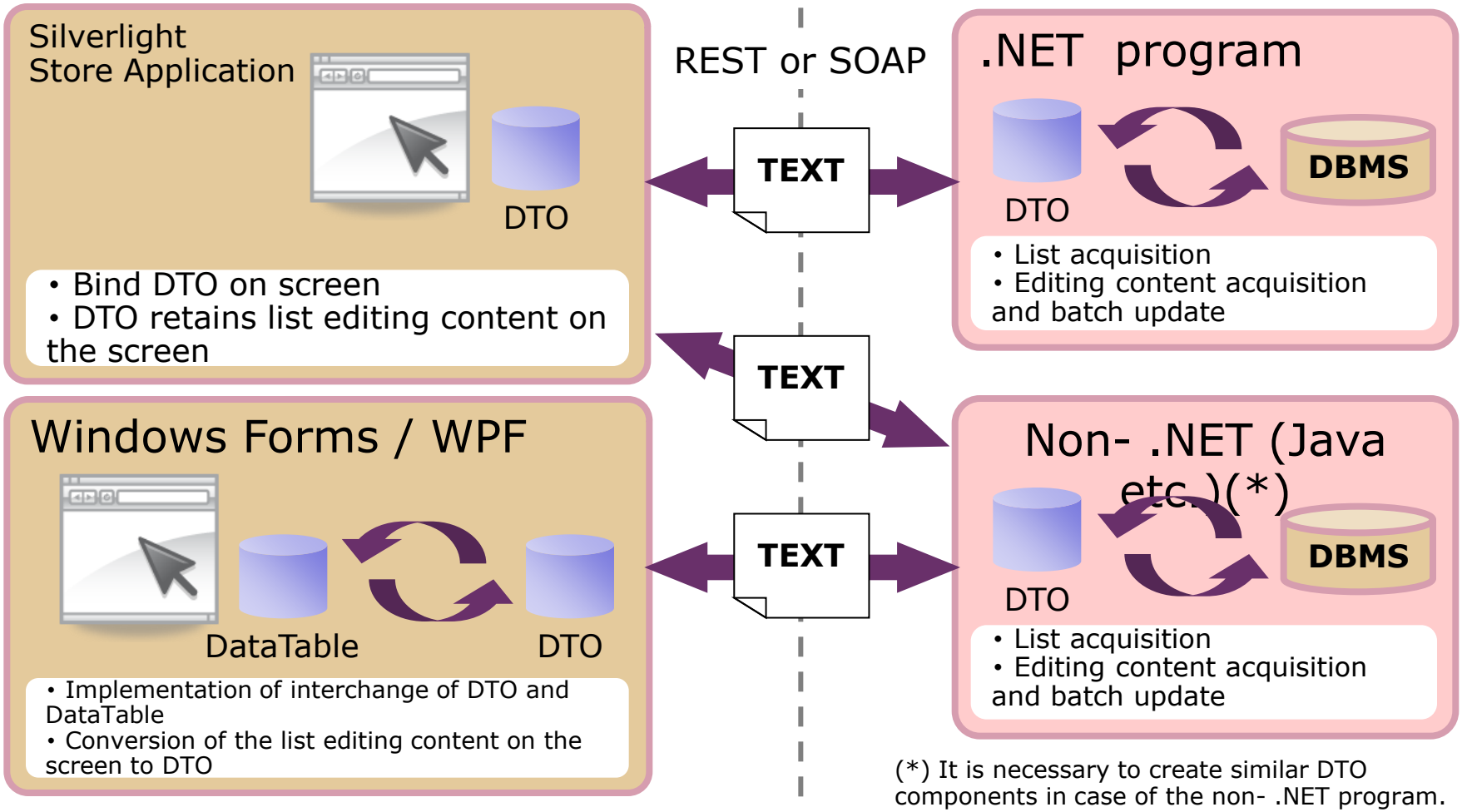


All types of documents

- **User guide**
- **Tutorials**
- **Samples etc.**

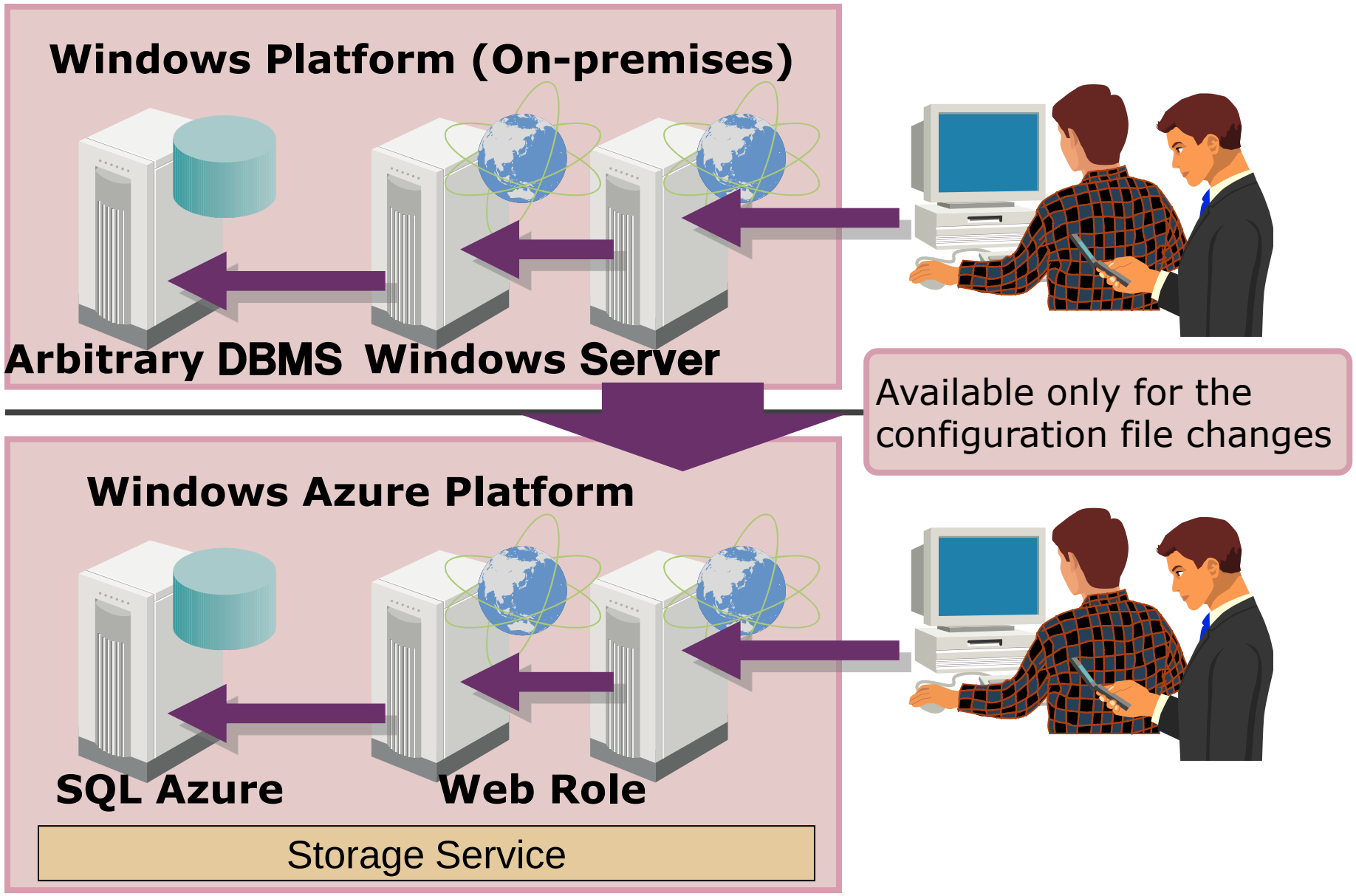
2.6 Silverlight Store Application Support

Even if Silverlight store application is selected as UI, it is possible to develop business application using the proven process methods, provided by "Touryo". Also, it is also possible to implement interoperation between .NET ⇔ non- .NET systems.



(*) It is necessary to create similar DTO components in case of the non- .NET program.

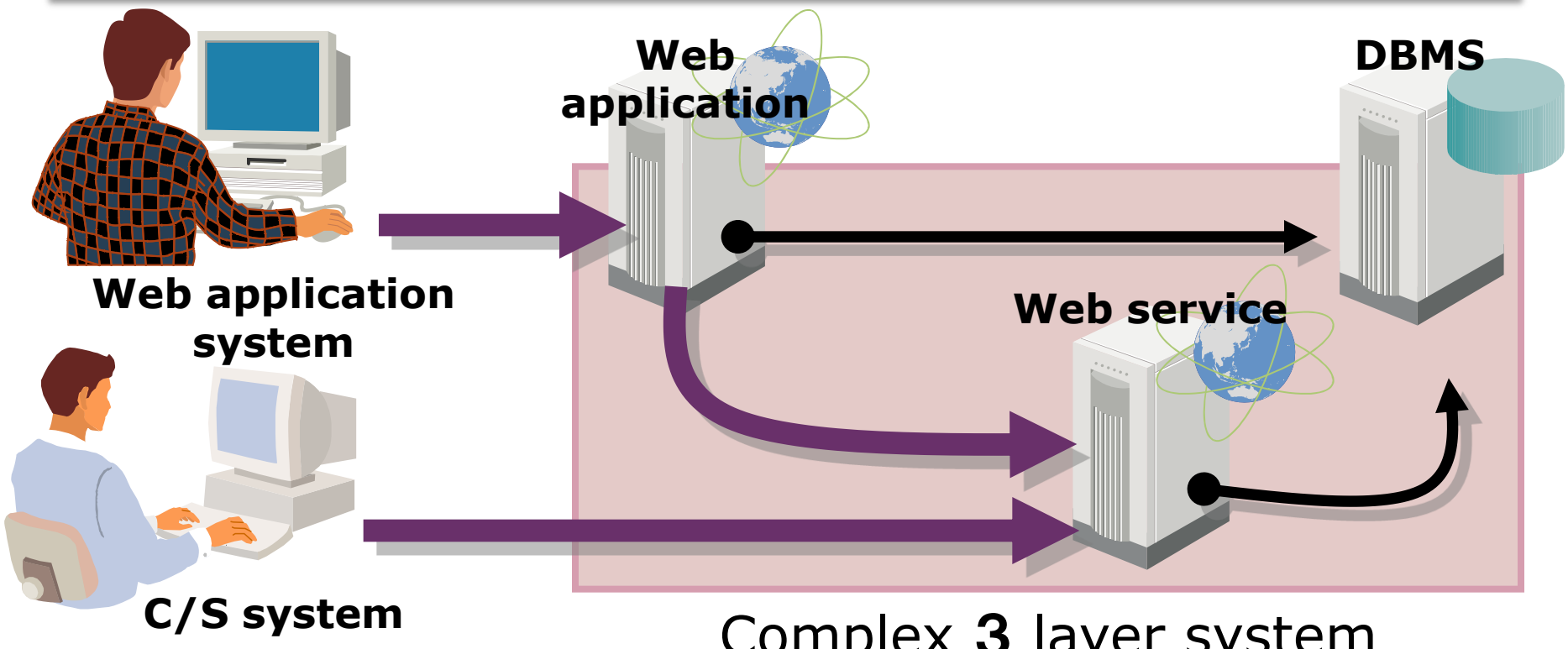
2.7 Windows Azure (Paas) Support



Contents

3.1 Communication Control Function – Summary

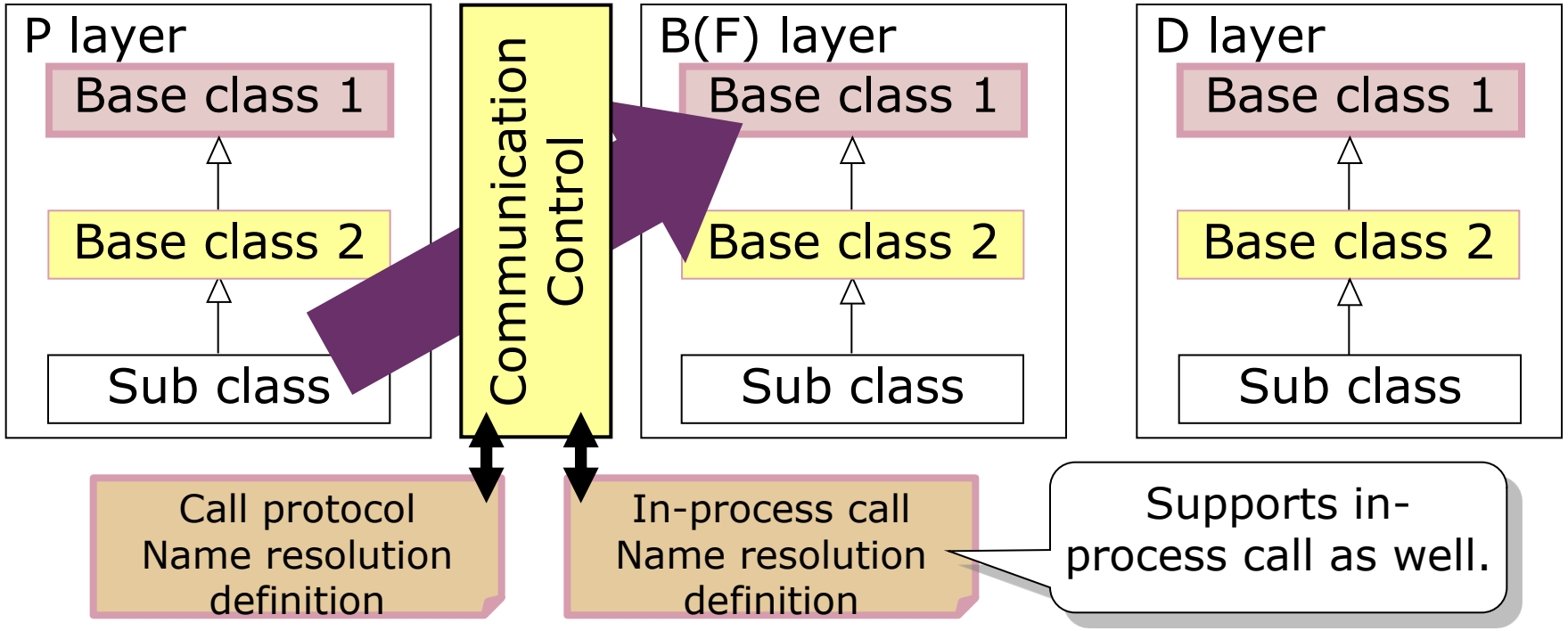
Communication control function facilitates the development of complex three-tier system.



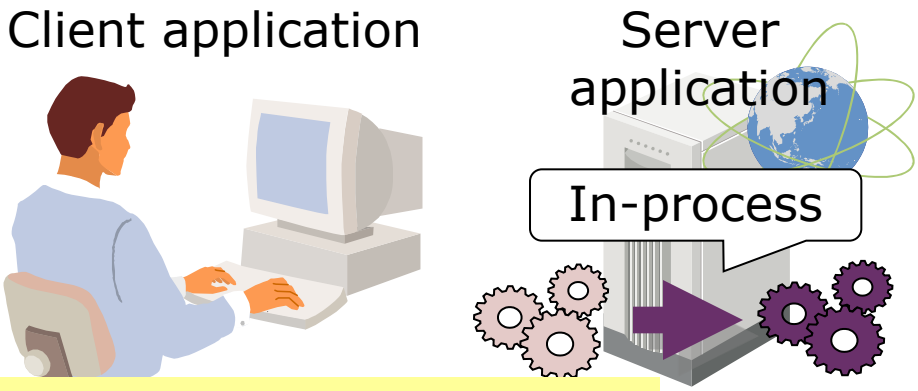
By communication control function to hide the cumbersome communication process, Developers can focus on the implementation of the business logic.

3.2 Communication Control Function – Add-in enabled Structure

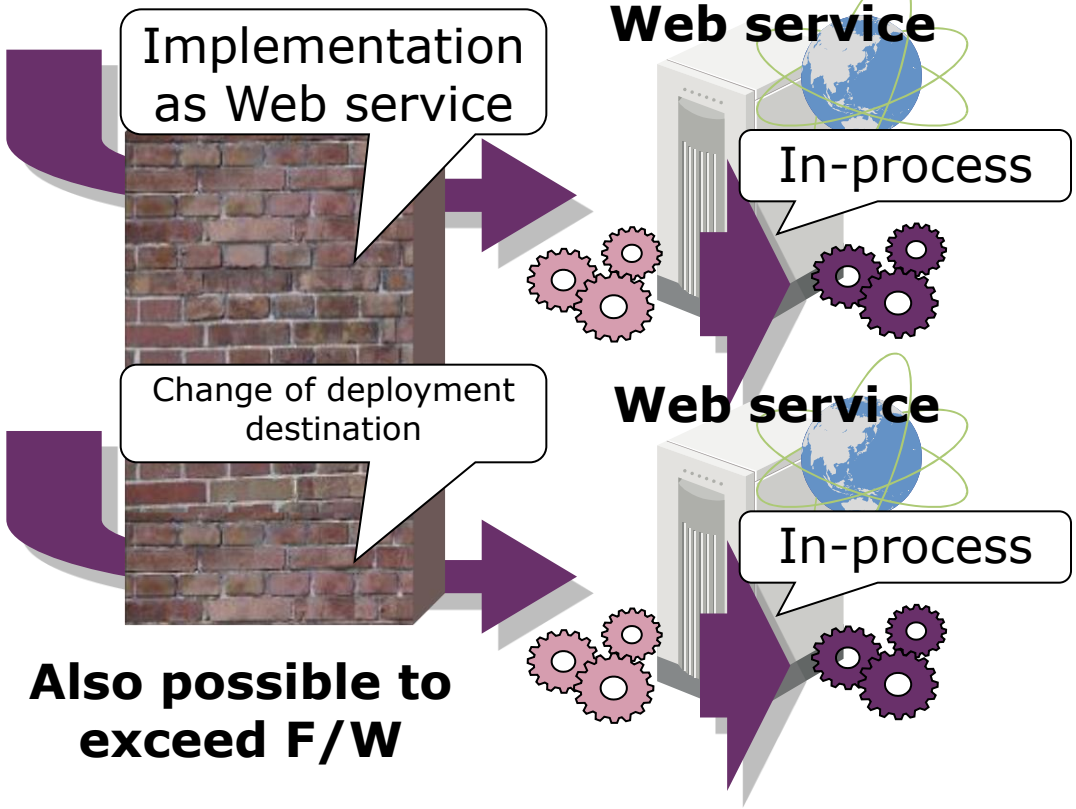
Communication control function can be added as an add-in to existing development. The distributed object type function is implemented using the Web service protocol.



3.3 Communication Control Function – Excellent Scalability



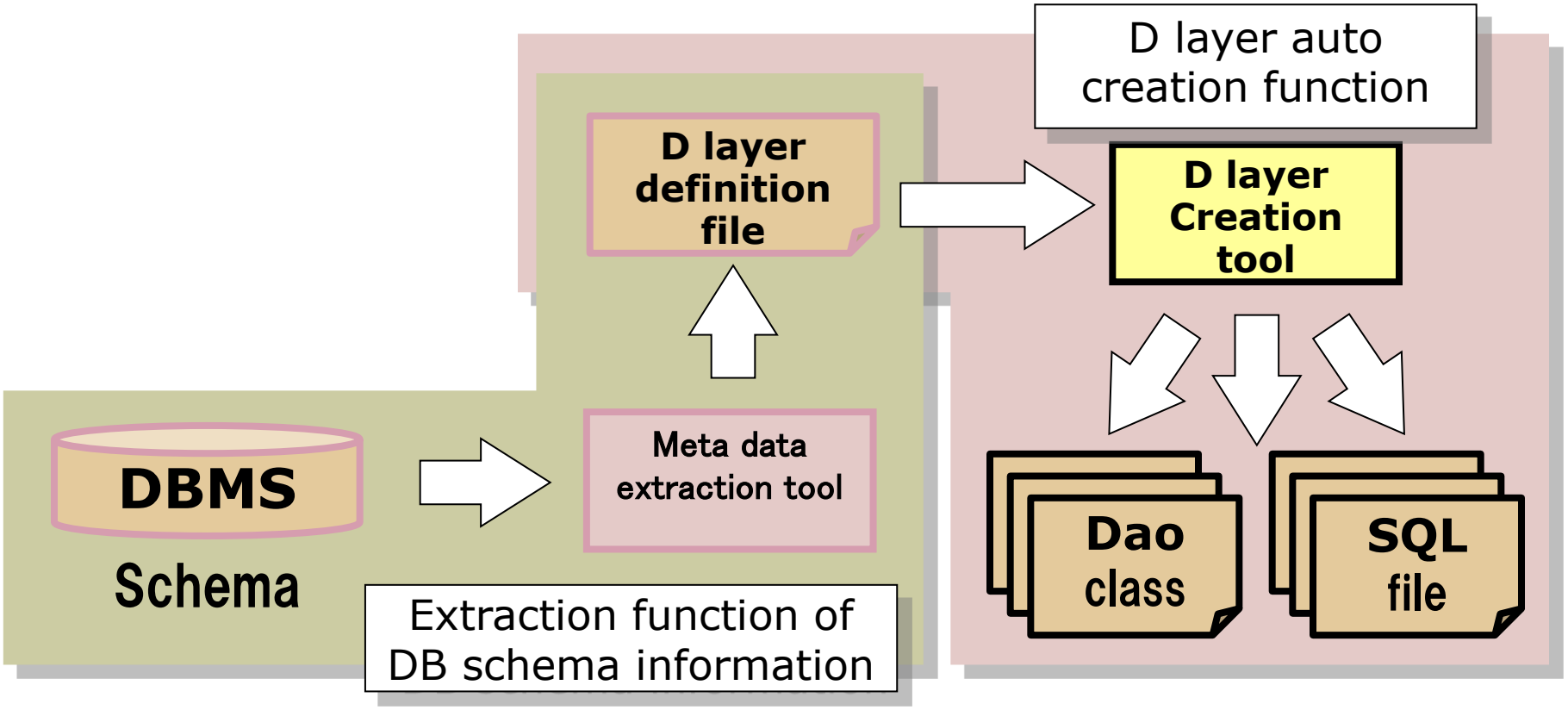
- **Location Transparency**
  Implementation of call switching through in-process/NW, change of call destination WAS by definition
- **Scale Transparency**
  Implementation of scale-out (vertical, horizontal dispersion)
- **Other Transparency**
  Also possible to connect with different kinds of development technology other than .NET



Contents

4. D layer Auto Creation Tool – Auto Creation

Automatically generates 100% Dao / SQL of CRUD processing for the table view with the schema information. Processing has been generated, it leverages the dynamic parameterized query below. In addition, time stamp optimism exclusive process of Web applications prone to leakage implementation will also be generated.



Contents

5.1 Dynamic Parameterized Query – Existing SQL

Previously, character string, character string concatenation, and the accompanying IF statement were scattered in the program that processes dynamic SQL.

Character string

```
'*********************************************************************************
'* SQL文作成
'*********************************************************************************
str_Sql = "SELECT COUNT(DISTINCT xxxxx) AS xxxxx FROM xxxxx"

If intXXXXX = 0 Then
  str_Work = " WHERE xxxxx = " & strXXXXX & " AND xxxxx LIKE '" & strXXXXX & "%'"
ElseIf intXXXXX = 2 Then
  str_Work = " WHERE xxxxx = " & strXXXXX & " AND xxxxx = '" & strXXXXX & "'"
ElseIf intXXXXX = 3 Then
  str_Work = " WHERE xxxxx = " & strXXXXX & " AND xxxxx = '" & strXXXXX & "'" & _
             " AND xxxxx IN (SELECT xxxxx FROM xxxxx WHERE xxxxx = " & strXXXXX & ")"
Else
  str_Work = " WHERE xxxxx = " & strXXXXX & " AND xxxxx LIKE '" & strXXXXX & "%'" & _
             " AND xxxxx IN (SELECT xxxxx FROM xxxxx WHERE xxxxx = " & strXXXXX & ")"
End If
```

Character string concatenation

IF statement

Program

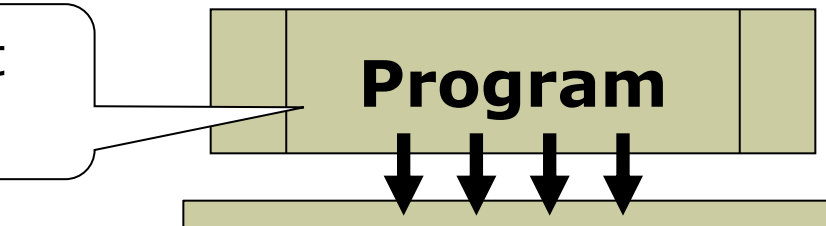
DBMS

Dynamic SQL is,
- Addition/deletion of WHERE, JOIN statement
- Addition/deletion of AND, OR operator
- Setup the parameter of IN statement so that it matches the no. of conditions
- Potential comprehension of vulnerability issue etc., as control is problematic and implementation is difficult.

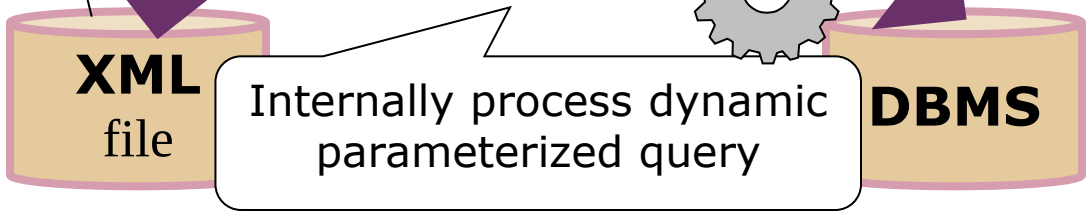
5.2 Dynamic Parameterized Query – Summary

Program implementation becomes easy due to Dynamic Parameterized Query Function, as there are no character string, character string concatenation, and the accompanying IF statements.

the parameter is set from the API only.




```
<?xml version="1.0" encoding="shift_jis" ?>
<ROOT>
  SELECT
    A, B, C, D
  FROM T
  <WHERE>
    WHERE
      <IF>AND A = @A</IF>
      <IF>AND A LIKE @A_LIKE</IF>
      <IF>AND B = @B</IF>
      <IF>AND B LIKE @B_LIKE</IF>
      <IF>AND C = @C</IF>
      <IF>AND C LIKE @C_LIKE</IF>
      <IF>AND D = @D</IF>
  </WHERE>
  ORDER BY <IF name="SEQUENCE">A<ELSE>B</ELSE></IF>
</ROOT>
```

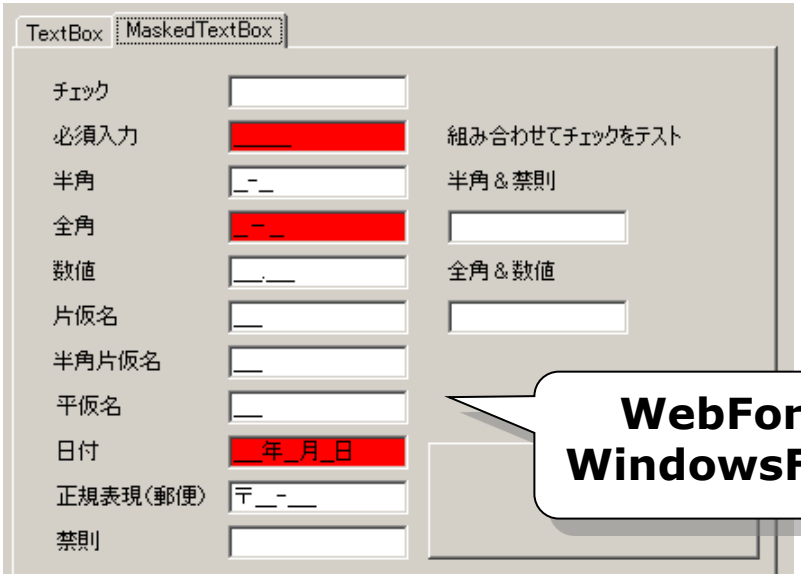


Dynamic parameterized query can be executed by parameter setting only, and developers are relieved from the control of complex character string concatenation processes.
Also, static parameterized query, which does not require description in XML, is also supported.

Contents

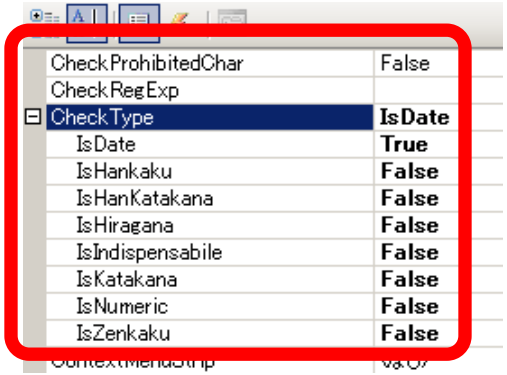
6.1 Custom Control and Validation

Easy implementation of item level check using custom control(WebForm / Windows Forms), and validation framework (WPF). Also, check conditions can be selected from VS designer and XAML based on attributes.



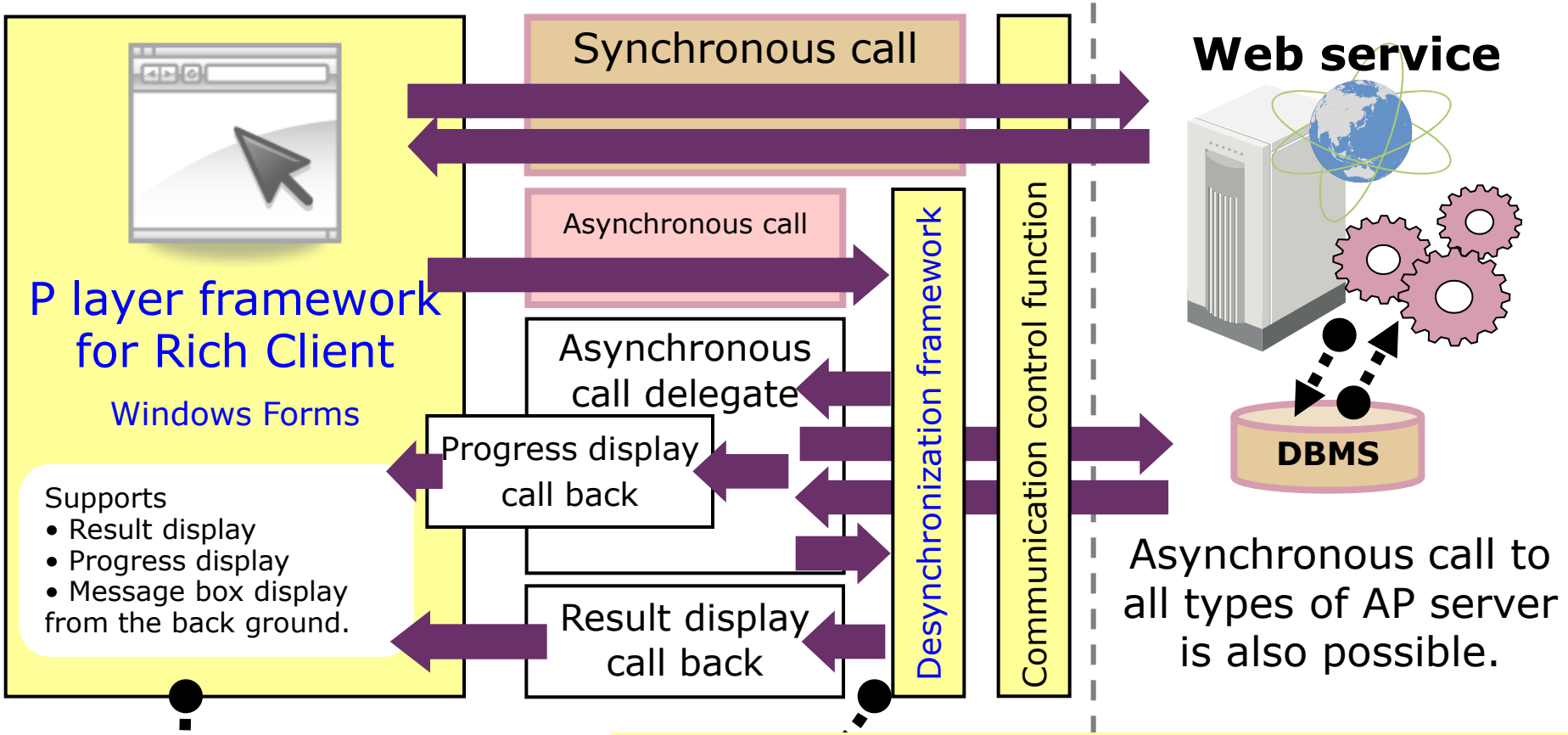
WebForm/ WindowsForms

WPF




```
<Label Height="23">カスタムのValidationRule（正の整数＋最大値設定可:200）
<TextBox Height="23" Margin="5" Name="textBox7"
         Validation.Error="textBox7_Error">
    <Binding Path="SourceProperty7"
             UpdateSourceTrigger="LostFocus"
        <Binding.ValidationRules>
            <!-- カスタムのValidationRule（最大値:200） -->
            <my:MyValidationRule Max="200"/>
        </Binding.ValidationRules>
    </
</TextBox>
```

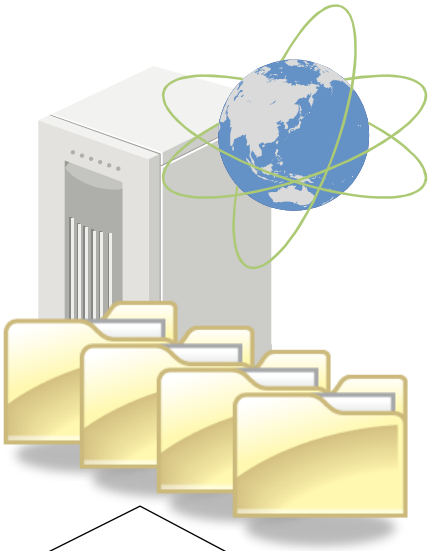
6.2 Rich Client supported P layer Framework



1.Migrated P layer framework for ASP.NET to rich client.
2.Implement asynchronous request, asynchronous polling using asynchronous framework.

6.3 Rich Client Web Deploy Tool

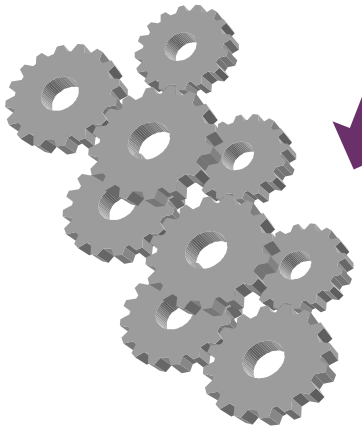
2. HEAD method
(Update confirmation)



1. Application start command via tool

3. GET method (ZIP acquisition)
(Supports differential download)

4. decompression/deployment/activation of downloaded ZIP file



Deployment manifesto file and ZIP file X n (ZIP file group with compressed program set)

Functions such as of ZIP file/deployment manifesto creation, MD5 hash check, silent mode execution is implemented in deploy tool.