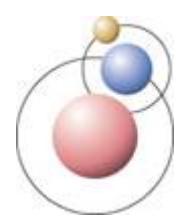


TERASOLUNA® Server Framework for .NET 2.1.0.1 アーキテクチャ説明書（Rich版）



株式会社NTTデータ



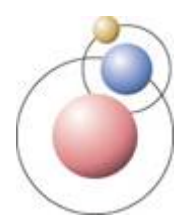


- 本ドキュメントを使用するにあたり、以下の規約に同意していただく必要があります。同意いただけない場合は、本ドキュメント及びその複製物の全てを直ちに消去又は破棄してください。
 1. 本ドキュメントの著作権及びその他一切の権利は、NTTデータあるいはNTTデータに権利を許諾する第三者に帰属します。
 2. 本ドキュメントの一部または全部を、自らが使用する目的において、複製、翻訳、翻案することができます。ただし本ページの規約全文、およびNTTデータの著作権表示を削除することはできません。
 3. 本ドキュメントの一部または全部を、自らが使用する目的において改変したり、本ドキュメントを用いた二次的著作物を作成することができます。ただし、「参考文献:TERASOLUNA Server Framework for .NET 2.1 アーキテクチャ説明書 (Rich版)」あるいは同等の表現を、作成したドキュメント及びその複製物に記載するものとします。
 4. 前2項によって作成したドキュメント及びその複製物を、無償の場合に限り、第三者へ提供することができます。
 5. NTTデータの書面による承諾を得ることなく、本規約に定められる条件を超えて、本ドキュメント及びその複製物を使用したり、本規約上の権利の全部又は一部を第三者に譲渡したりすることはできません。
 6. NTTデータは、本ドキュメントの内容の正確性、使用目的への適合性の保証、使用結果についての的確性や信頼性の保証、及び瑕疵担保義務も含め、直接、間接に被ったいかなる損害に対しても一切の責任を負いません。
 7. NTTデータは、本ドキュメントが第三者の著作権、その他如何なる権利も侵害しないことを保証しません。また、著作権、その他の権利侵害を直接又は間接の原因としてなされる如何なる請求(第三者との間の紛争を理由になされる請求を含む。)に関しても、NTTデータは一切の責任を負いません。
- 本ドキュメントで使用されている各社の会社名及びサービス名、商品名に関する登録商標および商標は、以下の通りです。
 - ◆ Microsoft、Visual Studio、Windows、.NET Frameworkは、米国Microsoft Corp.の米国及びその他の国における登録商標または商標です。
 - ◆ TERASOLUNAは、株式会社NTTデータの登録商標です。
 - ◆ その他の会社名、製品名は、各社の登録商標または商標です。



目次

- はじめに
- 動作環境
- Rich版アーキテクチャ概観
- 機能概要
- リクエスト制御概要
- 提供機能説明
 - ◆ プレゼンテーションレイヤ
 - WB-01 リクエストコントローラ機能
 - WB-02 ファイルアップロード機能
 - WB-03 ファイルダウンロード機能
 - CM-02 入力値検証機能
 - ◆ ビジネスレイヤ
 - CM-04 ビジネスロジック生成機能
 - ◆ データレイヤ
 - WC-02 SQL文管理機能
 - ◆ システム共通レイヤ
 - CM-01 メッセージ管理機能
 - CM-03 ログ出力機能



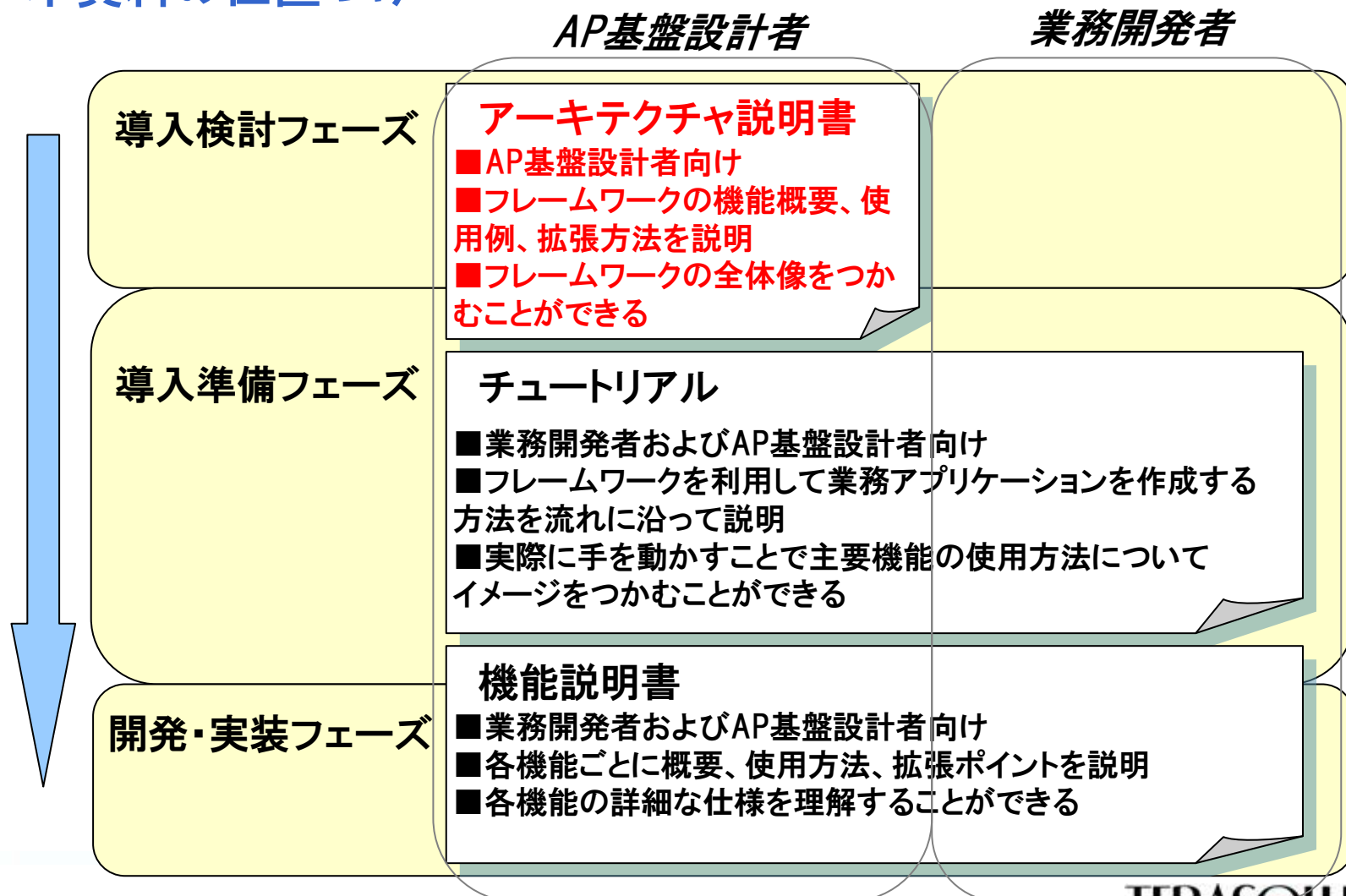
はじめに (1 / 4)

本資料は、「TERASOLUNA Server Framework for .NET 2.1 (Rich版)」の機能について解説した資料である。



はじめに (2/4)

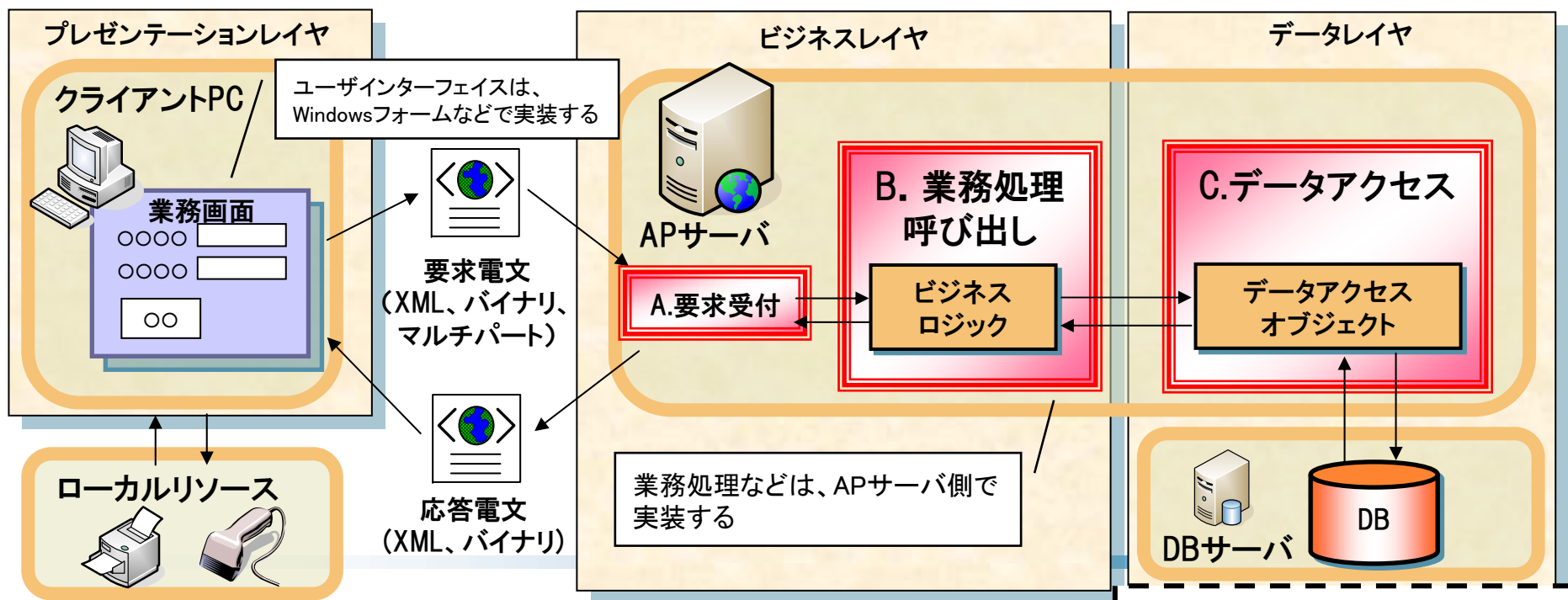
■ 本資料の位置づけ



はじめに (3/4)

■ TERASOLUNA Server Framework for .NETが想定する リッチクライアント型システム

- ◆ クライアントアプリケーションではプレゼンテーションレイヤに関する機能を実現する
 - ユーザインターフェイス
 - バーコードリーダやOCRなどの、クライアントリソースに依存した機能など
- ◆ サーバアプリケーションではビジネスレイヤ、データレイヤの機能を実現する
 - ビジネスレイヤに配置される業務処理、データソースへのアクセスなど

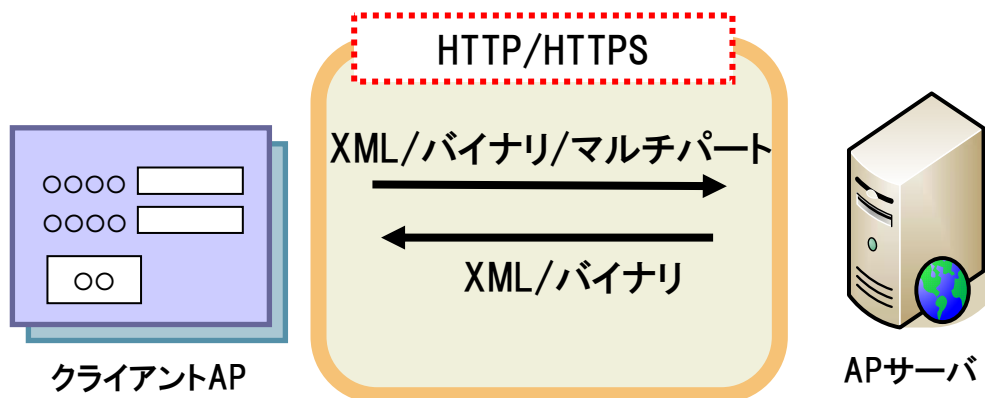




はじめに (4/4)

■ サーバ・クライアント間の通信方式の特徴

- ◆ クライアント・サーバ間は、HTTPまたはHTTPSでの通信とする
- ◆ HTTPボディ部の形式はXML形式、バイナリ形式、マルチパート形式の3パターンを前提としている
 - XML通信
 - － 要求がXML形式、応答がXML形式
 - ファイルアップロード
 - － 要求がマルチパート形式、応答がXML形式
 - － 要求がバイナリ形式、応答がXML形式
 - ファイルダウンロード
 - － 要求がXML形式、応答がバイナリ形式
- ◆ サーバはセッション状態を保持しない





動作環境

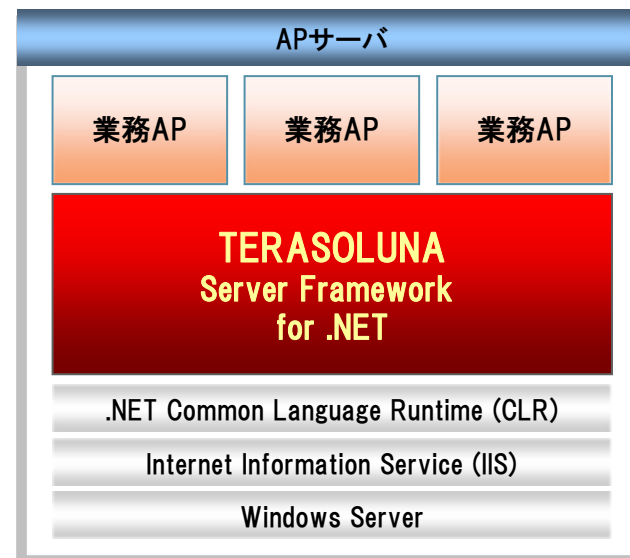
■ 動作確認環境

◆ Webサーバ

- Windows Server 2003 R2 SP2 上の IIS6.0

◆ .NET Framework

- .NET Framework 2.0
- .NET Framework 2.0 SP1



.NET Framework 3.0、3.5では動作確認を行っていない。

ただし、マイクロソフト社から .NET Framework 3.0 は .NET Framework 2.0 との、

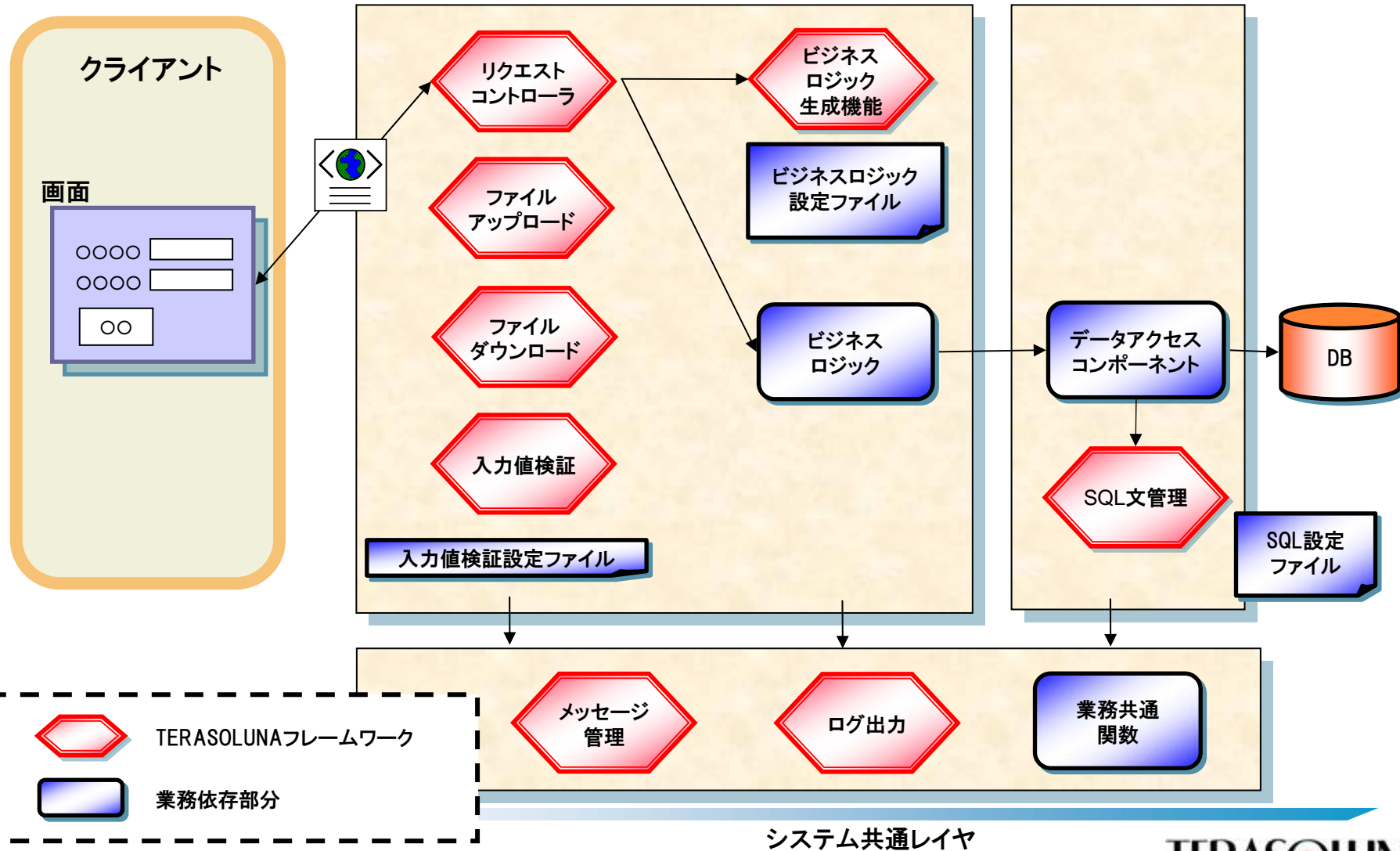
.NET Framework 3.5 は .NET Framework 2.0 SP1 との互換性が保証されている。

Rich版アーキテクチャ概観

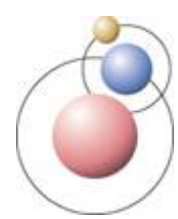
プレゼンテーションレイヤ

ビジネスレイヤ

データレイヤ



システム共通レイヤ



機能概要 (1/2)

■ ビジネスレイヤ

◆ WB-01 リクエストコントローラ機能

- クライアントからの要求に対して、サーバで業務処理を実行し、結果をクライアントへ返却する機能を提供する
 - クライアントから要求されたXML形式のメッセージを解析する
 - 解析した要求を入力として、業務処理を実行する
 - クライアントへ業務処理結果をXML形式で返却する

◆ WB-02 ファイルアップロード機能

- クライアントからアップロードされたマルチパート形式のデータやファイル(バイナリ)を受け付ける機能を提供する

◆ WB-03 ファイルダウンロード機能

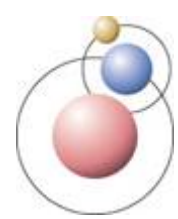
- 業務処理から返却されたファイル(バイナリ)を、クライアントへ返却する機能を提供する

◆ CM-02 入力値検証機能

- 業務処理に対する入力データを検証する機能を提供する

◆ CM-04 ビジネスロジック生成機能

- ビジネスロジッククラスのインスタンスを生成する機能を提供する



機能概要 (2/2)

■ データレイヤ

◆ WC-02 SQL文管理機能

- SQL文設定ファイルからSQL文を取得する機能を提供する

■ システム共通レイヤ

◆ CM-01 メッセージ管理機能

- アプリケーションで扱うメッセージに対し、統一的にアクセスする仕組みを提供する

◆ CM-03 ログ出力機能

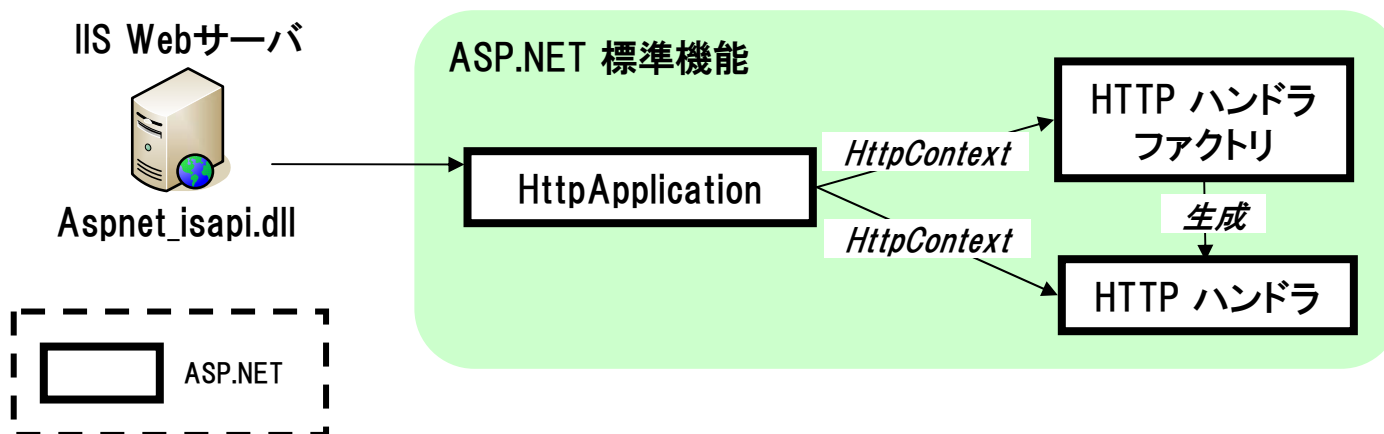
- アプリケーションで統一的にログを出力する仕組みを提供する



リクエスト制御概要 (1/2)

■ASP.NETランタイムのリクエスト制御の解説 (IIS6.0の場合)

- ◆ IIS Webサーバが受け付けた.aspxへの要求は、Aspnet_isapi.dll(ISAPIフィルタ)を通して、ASP.NETランタイムが制御する。このとき、HTTPハンドラファクトリが、実際にクライアントからの要求を処理するHTTPハンドラを生成する。HTTPハンドラは、要求電文の解析、応答電文の作成などリクエストに応じた個別処理を実行する。

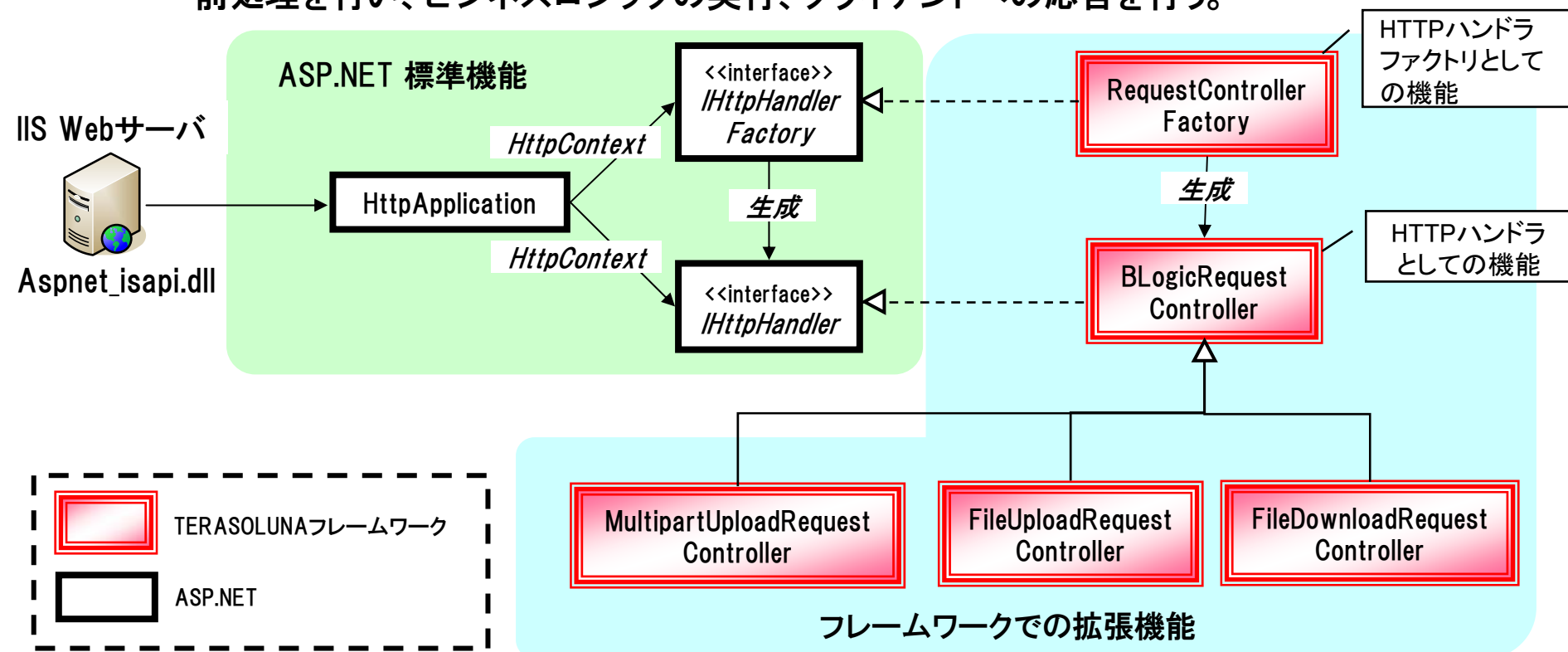


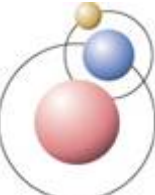
参考: HTTP ハンドラを使用したカスタム ASP.NET 処理
([http://msdn2.microsoft.com/ja-jp/library/5c67a8bd\(VS.80\).aspx](http://msdn2.microsoft.com/ja-jp/library/5c67a8bd(VS.80).aspx))

リクエスト制御概要 (2/2)

■ASP.NETランタイムとフレームワークとの位置づけ

- TERASOLUNAでは、HTTPハンドラファクトリとして、リクエストコントローラファクトリを提供する。リクエストコントローラファクトリは、リクエスト制御を行うリクエストコントローラを生成する。リクエストコントローラはクライアントからのビジネスロジック要求に対して必要な前処理を行い、ビジネスロジックの実行、クライアントへの応答を行う。



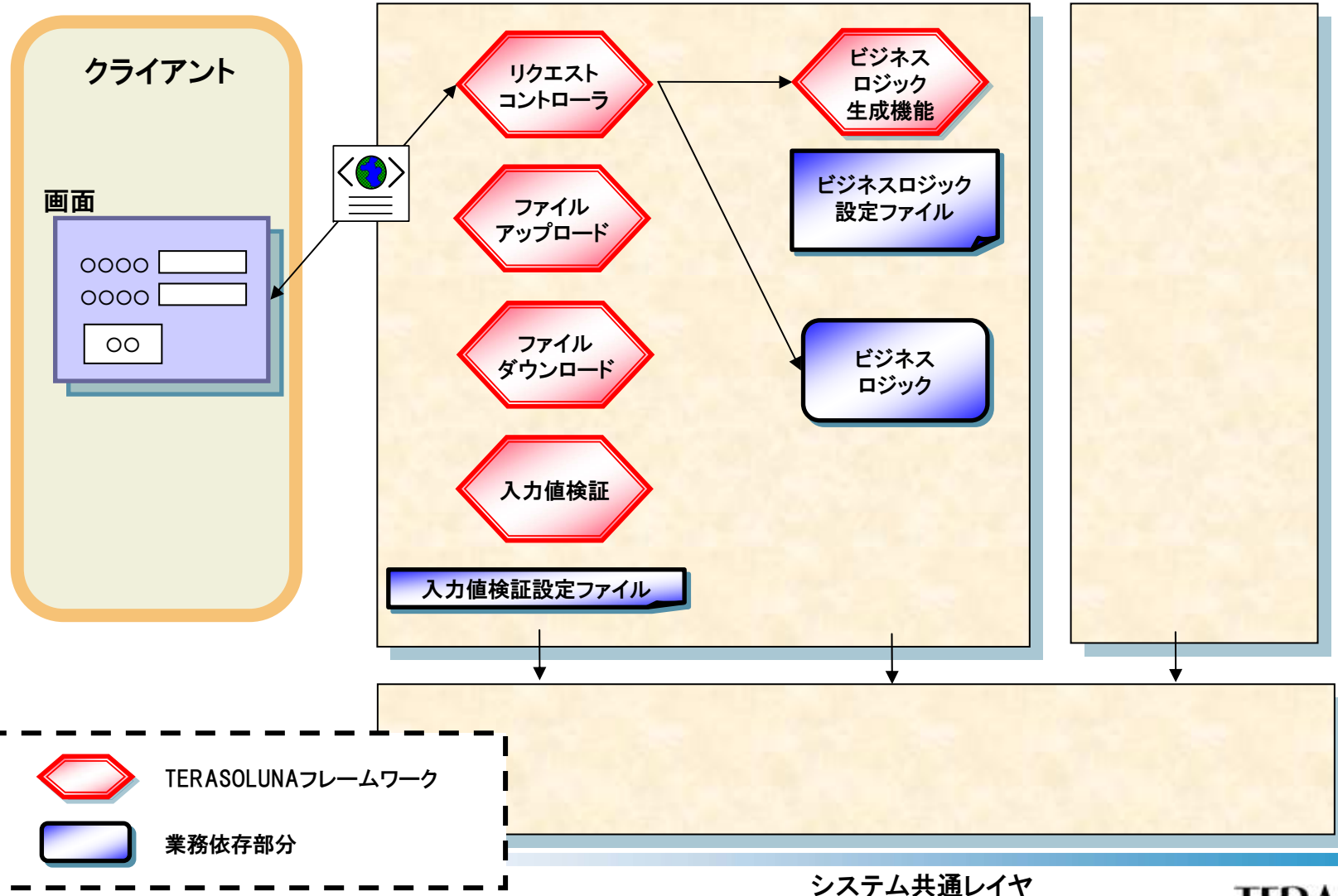


ビジネスレイヤ 機能説明

プレゼンテーションレイヤ

ビジネスレイヤ

データレイヤ



システム共通レイヤ



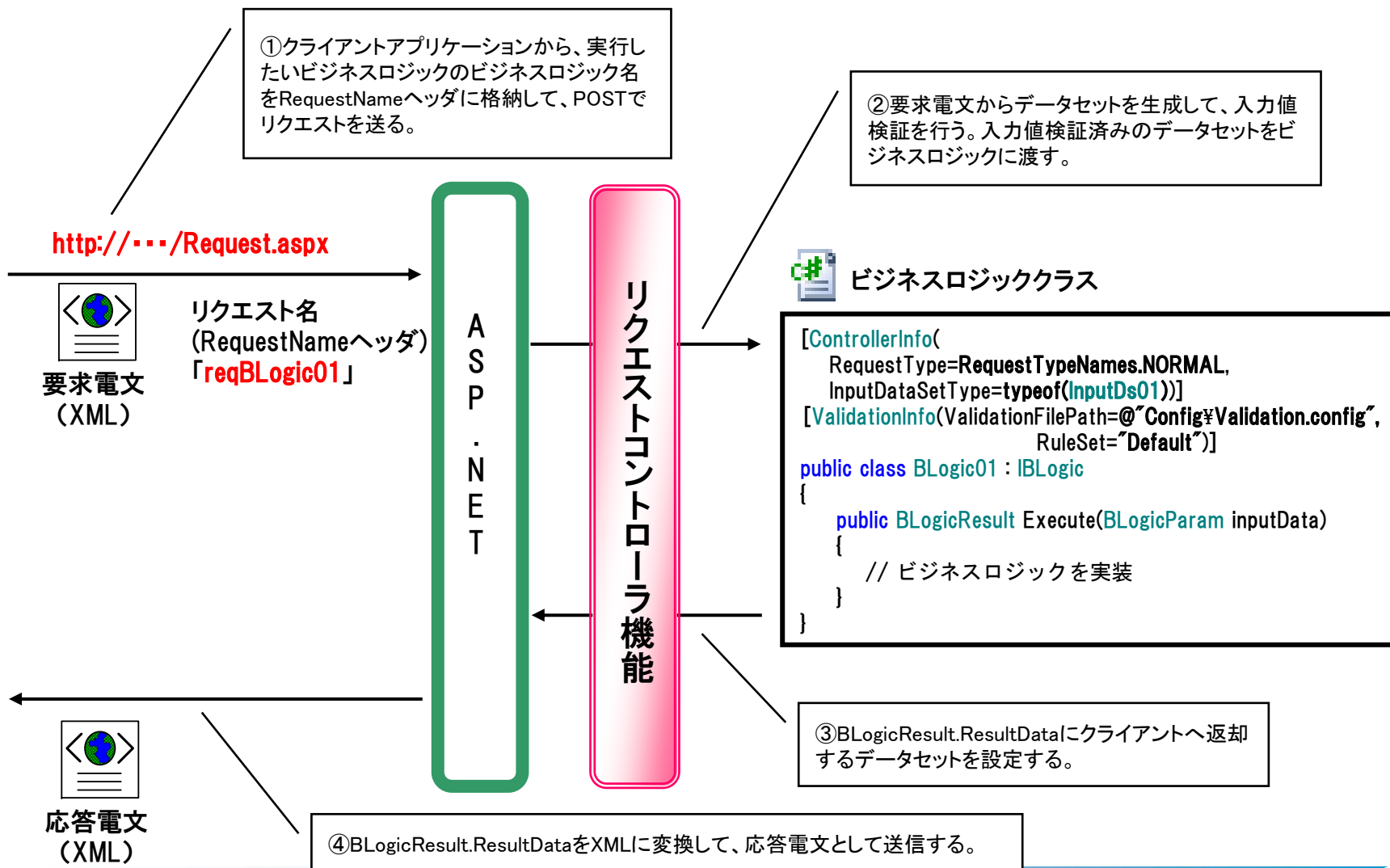


WB-01 リクエストコントローラ機能 – 概要

■ 機能概要

- ◆ クライアントからの要求に対して、サーバで業務処理を実行し、結果をクライアントへ返却する機能を提供する
 - クライアントから要求されたXML形式のメッセージを解析する
 - 解析した要求を入力として、業務処理を実行する
 - クライアントへ業務処理結果をXML形式で返却する

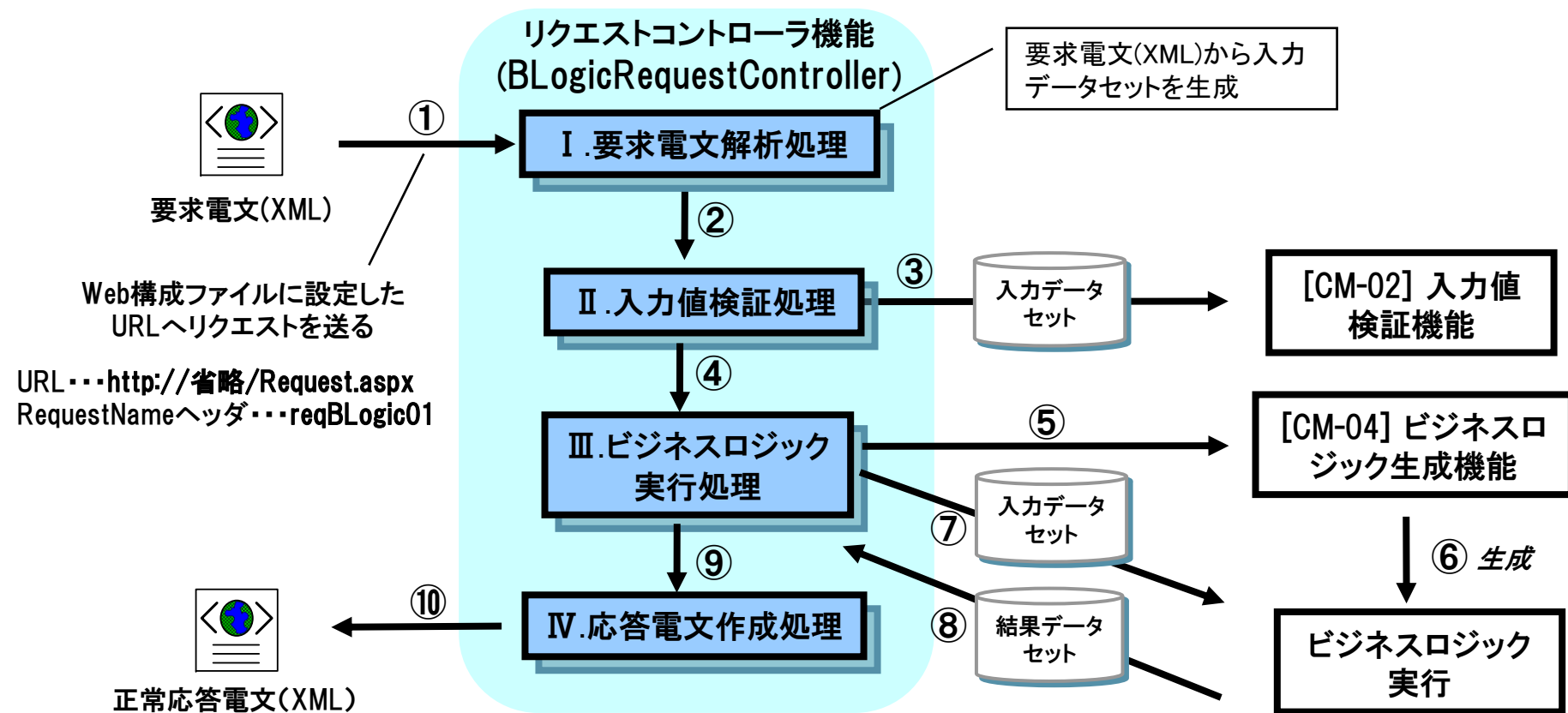
WB-01 リクエストコントローラ機能 – 動作イメージ (1/2)





WB-01 リクエストコントローラ機能 – 動作イメージ (2/2)

■ リクエストコントローラ機能の正常系動作詳細イメージ





WB-01 リクエストコントローラ機能 – 使用方法 (1/6)

■ Web構成ファイルの設定(1)

◆ リクエストコントローラを利用するための設定をする

```
<!--リクエストコントローラの設定-->
```

```
<system.web>
```

```
<httpHandlers>
```

```
<add verb="POST" path="Request.aspx"
```

```
type="TERASOLUNA.Fw.Web.Controller.RequestControllerFactory, TERASOLUNA.Fw.Web"/>
```

```
</httpHandlers>
```

```
</system.web>
```

リクエストを受け付けるURLを指定。
任意のURLでよい。



WB-01 リクエストコントローラ機能 – 使用方法 (2/6)

■ Web構成ファイルの設定(2)

◆ 利用するビジネスロジック設定ファイルのパスを設定する

```
<!--ビジネスロジック設定ファイルのパスの設定-->
<configSections>
  <section name="blogicConfiguration"
    type="TERASOLUNA.Fw.Common.Configuration.BLogic.BLogicConfigurationSection, TERASOLUNA.Fw.Common"/>
</configSections>
<blogicConfiguration>
  <files>
    <file path="Config¥BLogicConfig.config"/>
  </files>
</blogicConfiguration>
```



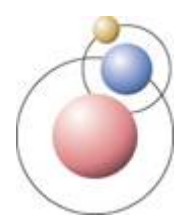
WB-01 リクエストコントローラ機能 – 使用方法 (3/6)

■ ビジネスロジック設定ファイルの設定

◆ ビジネスロジック名とビジネスロジックの型を設定する

```
<!--ビジネスロジック設定ファイル-->
<?xml version="1.0" encoding="utf-8" ?>
<blogicConfiguration xmlns="http://www.terasoluna.jp/schema/BLogicSchema.xsd">
  <blogic name="reqBLogic01" type="Business.BLogic.BLogic01, Business" />
  <blogic name="reqBLogic02" type="Business.BLogic.BLogic02, Business" />
  <blogic name="reqBLogic03" type="Business.BLogic.BLogic03, Business" />
</blogicConfiguration>
```

ビジネスロジック名はクライアントからのリクエストの
RequestNameヘッダの値に対応する



WB-01 リクエストコントローラ機能 – 使用方法 (4/6)

■ ビジネスロジックの設定

- ◆ ビジネスロジッククラスにリクエストコントローラと入力値検証に関する属性情報を指定する
 - ControllerInfo
 - RequestType...使用するリクエストコントローラを特定する文字列
「RequestTypeNames.NORMAL」を設定する
 - InputDataSetType...Executeメソッドの引数に渡ってくるデータセットの型
 - ValidationInfo(入力値検証を有効とする場合のみ)
 - ValidationFilePath...入力値検証設定ファイルのパス
 - RuleSet...入力値検証で利用するルールセット

◆ 設定例

```
[ControllerInfo(RequestType=RequestTypeNames.NORMAL,  
                 InputDataSetType=typeof(InputDs01))]  
[ValidationInfo(ValidationFilePath=@"Config¥Validation.config",  
                RuleSet="Default")]  
public class BLogic01 : ILogic  
{  
    public BLogicResult Execute(BLogicParam inputData)  
    {  
        // ビジネスロジックを実装  
    }  
}
```



WB-01 リクエストコントローラ機能 – 使用方法 (5/6)

■ビジネスロジックに設定する属性情報の種類

属性	パラメータ	概要
ControllerInfo	RequestType	要求の解決に利用するリクエストコントローラの識別子。 なお、設定を省略すると既定値が使われる。 ●フレームワークが提供するHTTPハンドラ ・Normal(XML送受信) BLogicRequestControllerを利用 (既定値) ・Multipart(マルチパートアップロード) MultipartUploadRequestControllerを利用 ・Upload(ファイルアップロード) FileUploadRequestControllerを利用 ・Download(ファイルダウンロード) FileDownloadRequestControllerを利用
	InputDataSetType	ビジネスロジックの入力データセットとなる、型付データセットの型情報。
ValidationInfo	ValidationFilePath	入力値検証に利用する入力値検証設定ファイルのパス。 Binディレクトリからの相対パスを指定する。設定しない場合は入力値検証は行われない。
	RuleSet	入力値検証に利用するルールセット。設定しない場合は、“Default”となる。



WB-01 リクエストコントローラ機能 – 使用方法 (6/6)

◆ 処理結果と応答電文の関係

	条件	HTTPレスポンス	
		exceptionヘッダ	ボディ
ビジネスロジック 正常終了	ビジネスロジックExecuteメソッドから返却される、BLogicResult.Errorsの件数が0件だった場合	なし	ビジネスロジック出力データセット (BLogicResult.ResultData)をXML形式でクライアントへ返す。
ビジネスロジック 業務エラー	ビジネスロジックExecuteメソッドから返却される、BLogicResult.Errorsの件数が1件以上だった場合	BLogicResult.ResultString	エラーメッセージ情報(BLogicResult.Errors)を、XML形式でクライアントへ返す。応答電文(XML)のフォーマットは、機能説明書「WB-01 リクエストコントローラ機能」を参照のこと。
入力値検証エ ラー	入力値検証終了後、ValidationResult.Errorsの件数が1件以上だった場合	" validateException "	入力値検証エラーメッセージ情報を、XML形式でクライアントへ返す。応答電文(XML)のフォーマットは、機能説明書「WB-01 リクエストコントローラ機能」を参照のこと。
システムエラー	クライアントからの要求が不正、または業務処理で捕捉されない例外が発生した場合	" exception "	発生例外のメッセージと、例外に対応付けられたエラーコードを、XML形式でクライアントへ返す。応答電文(XML)のフォーマットとエラーコードのマッピング方法は、機能説明書「WB-01 リクエストコントローラ機能」を参照のこと。



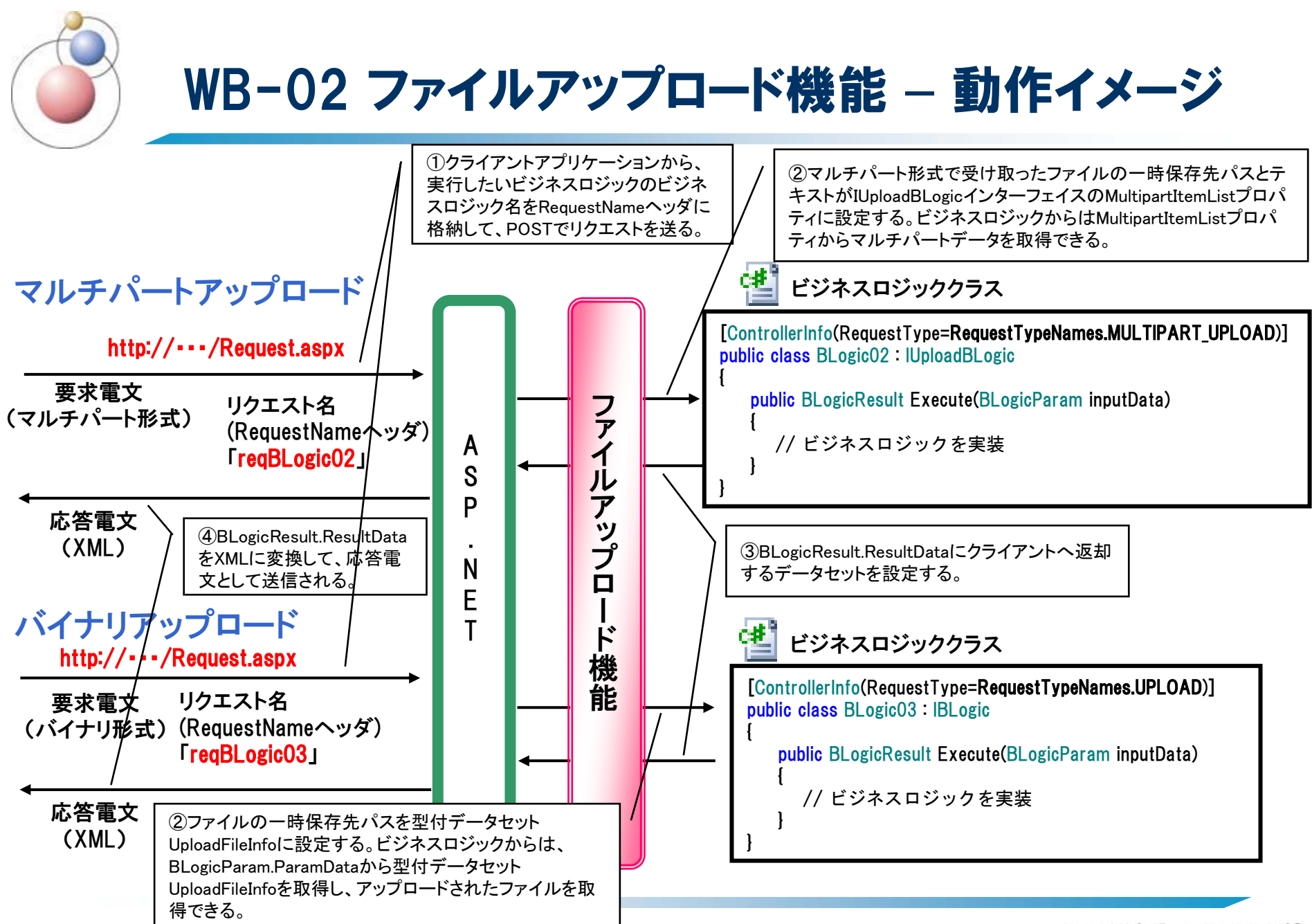
WB-02 ファイルアップロード機能 - 概要

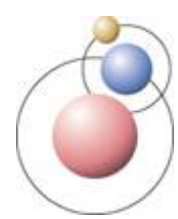
■ 機能概要

- ◆ クライアントからのファイルアップロード要求に対し、ビジネスロジックの呼び出し、レスポンスを送信する機能を提供する
 - アップロードの形式として、マルチパート形式とバイナリ形式をサポートする

	アップロードデータのビジネスロジックへの通知方法
マルチパート形式	マルチパート形式のデータを解析し、ファイルの一時保存パスとテキストデータをビジネスロジックへ通知する
バイナリ形式	ファイルの一時保存パスをビジネスロジックへ通知する

WB-02 ファイルアップロード機能 – 動作イメージ





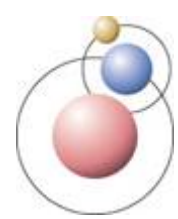
WB-02 ファイルアップロード機能 – 使用方法 (1/3)

■ Web構成ファイルの設定

- ◆ 「WB-01 リクエストコントローラ機能」と同様

■ ビジネスロジック設定ファイルの設定

- 「WB-01 リクエストコントローラ機能」と同様



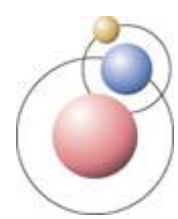
WB-02 ファイルアップロード機能 – 使用方法 (2/3)

■ マルチパートアップロード用ビジネスロジックの設定

- ◆ ビジネスロジックは IUploadBLogic を実装する
- ◆ ビジネスロジッククラスにリクエストコントローラに関する属性情報を指定する
 - ControllerInfo
 - RequestType...使用するリクエストコントローラを特定する文字列
「RequestTypeNames.MULTIPART_UPLOAD」を設定する
- ◆ 設定例
 - MultipartItemListプロパティからマルチパートデータを取得できる

```
[ControllerInfo(RequestType=RequestTypeNames.MULTIPART_UPLOAD)]
public class BLogic02 : IUploadBLogic
{
    public BLogicResult Execute(BLogicParam inputData)
    {
        foreach (IMultipartItem multipartItem in MultipartItemList)
        {
            // マルチパートデータの取得
        }

        // ビジネスロジックを実装
    }
}
```



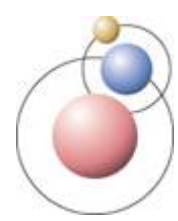
WB-02 ファイルアップロード機能 – 使用方法 (3/3)

■ バイナリアップロード用ビジネスロジックの設定

- ◆ ビジネスロジッククラスにリクエストコントローラに関する属性情報を指定する
 - ControllerInfo
 - RequestType...使用するリクエストコントローラを特定する文字列
「RequestTypeNames.UPLOAD」を設定する
- ◆ 設定例
 - BLogicParam.ParamDataに設定されている型付データセットUploadFileInfoから、アップロードされたファイルのパスを取得できる

```
[ControllerInfo(RequestType=RequestTypeNames.UPLOAD)]
public class BLogic03 : IUploadBLogic
{
    public BLogicResult Execute(BLogicParam inputData)
    {
        // アップロードファイルパスを取得
        UploadFileInfo inputDs = (UploadFileInfo)inputData.ParamData;

        // ビジネスロジックを実装
    }
}
```

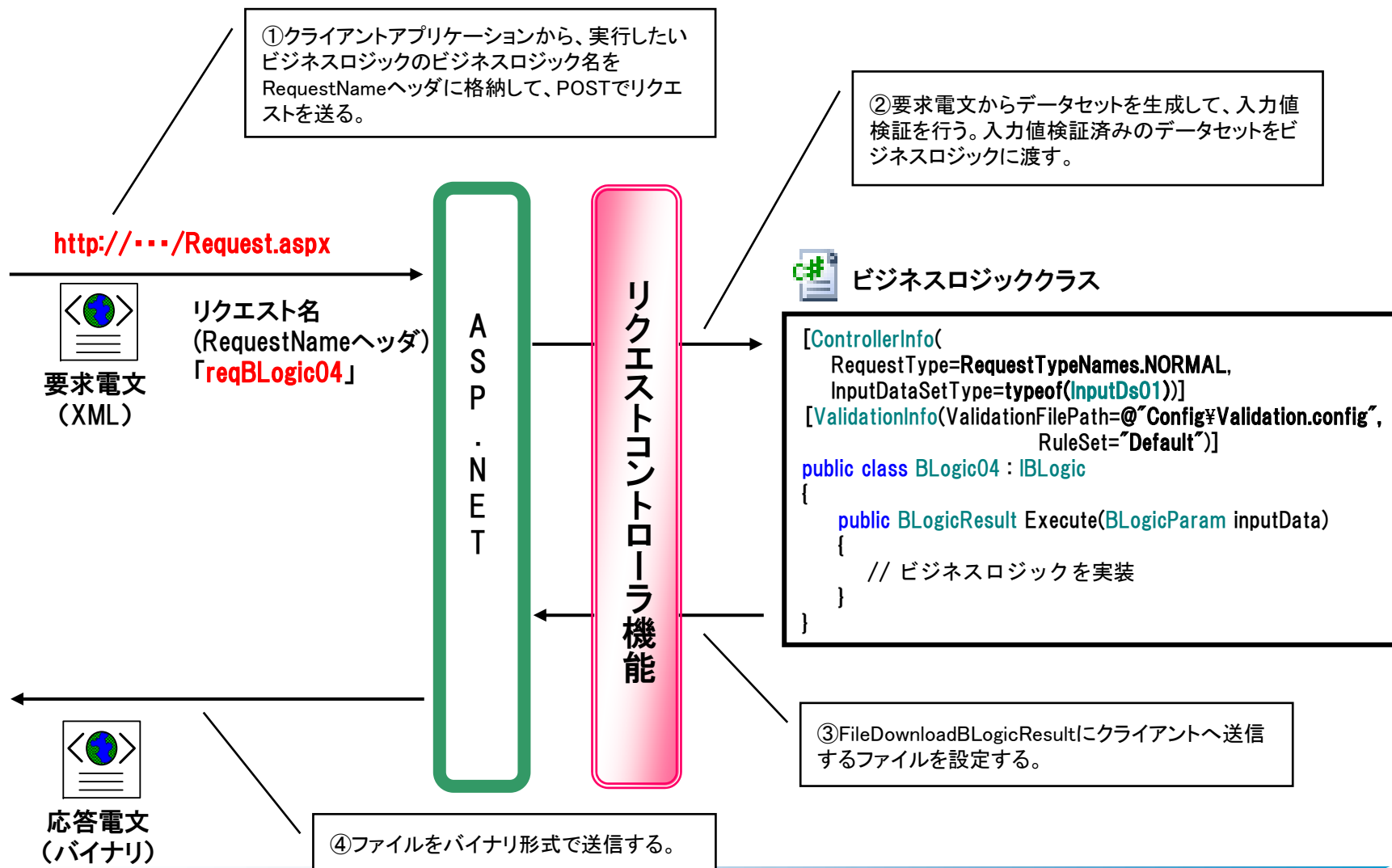


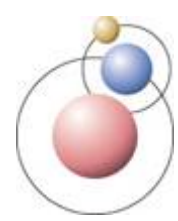
WB-03 ファイルダウンロード機能 - 概要

■ 機能概要

- ◆ クライアントからのビジネスロジック実行要求に対し、リクエストの検証、ビジネスロジックの呼び出し、クライアントへファイルを送信する機能を提供する

WB-03 ファイルダウンロード機能 – 動作イメージ





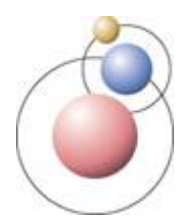
WB-03 ファイルダウンロード機能 – 使用方法 (1/2)

■ Web構成ファイルの設定

- ◆ 「WB-01 リクエストコントローラ機能」と同様

■ ビジネスロジック設定ファイルの設定

- 「WB-01 リクエストコントローラ機能」と同様



WB-03 ファイルダウンロード機能 – 使用方法 (2/2)

■ ファイルダウンロード用ビジネスロジックの設定

- ◆ ビジネスロジッククラスにリクエストコントローラに関する属性情報を指定する
 - ControllerInfo
 - RequestType...使用するリクエストコントローラを特定する文字列「RequestTypeNames.DOWNLOAD」を設定する
- ◆ 設定例
 - FileDownloadBLogicResultにクライアントへ送信するファイル情報を設定する

```
[ControllerInfo(RequestType=RequestTypeNames.DOWNLOAD)]  
public class BLogic04 : IUploadBLogic  
{  
    public BLogicResult Execute(BLogicParam inputData)  
    {  
        // ビジネスロジックの実装  
  
        // ダウンロードファイル情報を返却  
        return new FileDownloadBLogicResult(fileInfo);  
    }  
}
```




CM-02 入力値検証機能

■ 機能概要

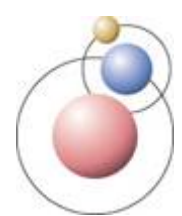
- ◆ リクエストコントローラ機能、ファイルダウンロード機能で要求電文(XML)の入力値検証を行う機能を提供する
 - ファイルアップロード機能では入力値検証は行われない
- ◆ 詳細は「TERASOLUNA Server/Client Framework for .NET 2.1 アーキテクチャ説明書(共通編)」を参照のこと



CM-04 ビジネスロジック生成機能

■ 機能概要

- ◆ ビジネスロジッククラスのインスタンスを生成する機能を提供する
- ◆ 詳細は「TERASOLUNA Server/Client Framework for .NET 2.1 アーキテクチャ説明書(共通編)」を参照のこと



データレイヤ 機能説明

プレゼンテーションレイヤ

ビジネスレイヤ

データレイヤ

クライアント

画面

○○○○

○○○○

○○

データアクセス
コンポーネント

SQL文管理

DB

SQL設定
ファイル

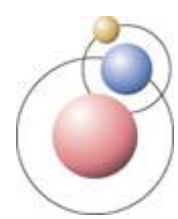


TERASOLUNAフレームワーク



業務依存部分

システム共通レイヤ



WC-02 SQL文管理機能

■機能概要

- ◆ SQL文設定ファイルからSQL文を取得する機能を提供する
 - SQL文のIDに対応するSQL文を返却する
 - SQL文を設定ファイルから取得することにより、再ビルドを行わずにSQL文を変更できる



WC-02 SQL文管理機能 – 使用方法 (1/2)

■ Web構成ファイルの設定

◆ 利用するSQL文設定ファイルのパスを設定する

```
<!--SQL文設定ファイルのパスの設定-->
<configSections>
  <section name="sqlConfiguration"
    type="TERASOLUNA.Fw.Web.Configuration.Sql.SqlConfigurationSection, TERASOLUNA.Fw.Web"/>
</configSections>
<sqlConfiguration>
  <files>
    <file path="Config¥SqlConfiguration.config" />
  </files>
</sqlConfiguration>
```



WC-02 SQL文管理機能 – 使用方法 (2/2)

■ SQL文設定ファイルの設定

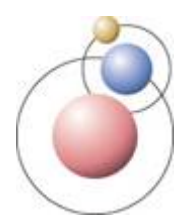
◆ SQL IDとSQL文を設定する

```
<!--SQL文設定ファイルの設定-->
<?xml version="1.0" encoding="utf-8" ?>
<sqlConfiguration xmlns="http://www.terasoluna.jp/schema/SqlSchema.xsd">
  <sql name="selectSample">
    <![CDATA[
      SELECT * FROM TABLE
    ]]>
  </sql>
</sqlConfiguration>
```

■ SQL文の取得

SQL文取得のコード例

```
string query = SqlConfiguration.GetSql("selectSample");
```



システム共通レイヤ 機能説明

プレゼンテーションレイヤ

ビジネスレイヤ

データレイヤ

クライアント

画面



TERASOLUNAフレームワーク



業務依存部分

メッセージ
管理

ログ出力

業務共通
関数

システム共通レイヤ

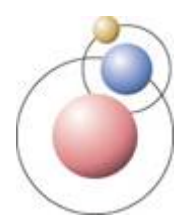
TERASOLUNA



CM-01 メッセージ管理機能

■ 機能概要

- ◆ アプリケーションで扱うメッセージに対し、統一的にアクセスする仕組みを提供する
- ◆ 詳細は「TERASOLUNA Server/Client Framework for .NET 2.1 アーキテクチャ説明書(共通編)」を参照のこと



CM-03 ログ出力機能

■ 機能概要

- ◆ アプリケーションで統一的にログを出力する仕組みを提供する
- ◆ 詳細は「TERASOLUNA Server/Client Framework for .NET 2.1 アーキテクチャ説明書(共通編)」を参照のこと