



# EZT開発フレームワーク ご紹介資料

エバーゼスト株式会社



## 1. WebAPIを利用したEZTフレームワークのご紹介

昨今では、さまざまな、端末がネットワークに接続される環境が整ってきており、PCクライアントだけではなく、携帯・スマホ・各種センサ端末がネットワークに接続されています。

今後は、これらの端末を利用して様々なサービスを提供していく必要が出てくると考えられます。

弊社では、これらの環境に対応した、様々なサービスを構築する為の、軽量のフレームワークを開発し提供させていただきます。

### 【特徴】

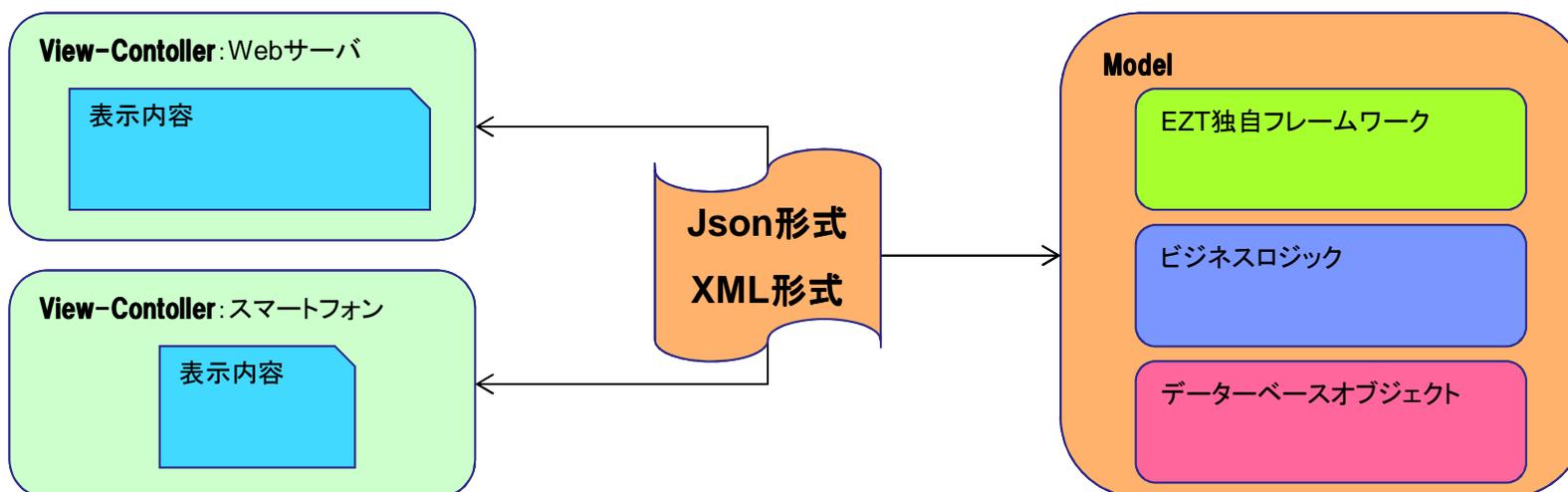
- WebAPI(\*1)方式による多様なクライアント端末の接続できます。
- 弊社開発のベースクラスから、お客様のシステム要件に合わせたフレームワークを有償で提供します。
  - Webデザインパターン(\*2)を意識しない開発が可能な為、経験が浅い要員でも対応可能です。
- お客様では、ビジネスロジックのみを開発することでシステム構築ができます。
- 機能の拡張はWebAPIに新たなバージョンのAPIを増やすことにより、バージョンの古い端末から最新の端末までサポートできます。
- シンプルなフレームワーク構造の為、軽量かつ軽快に動作します。

※1 WebAPIとは「Webサイトなどの開発を効率的に行うための技術」

※2 オブジェクト指向開発で再利用性の高い設計をするためのパターン(雛形)

## 2. EZT独自フレームワーク構成イメージ

- EZT独自フレームワークはMVCモデルに準拠しています
  - MVCに準拠しているため、各機能の組み合わせにより、高効率化を実現します。
- MVCモデルとは
  - 「ソフトウェアの設計モデルの一つで、処理の中核「Model」、表示／出力の「View」、入力を受取り内容に応じてViewとModelを制御する「Controller」の3要素システムを実装する方式」です。



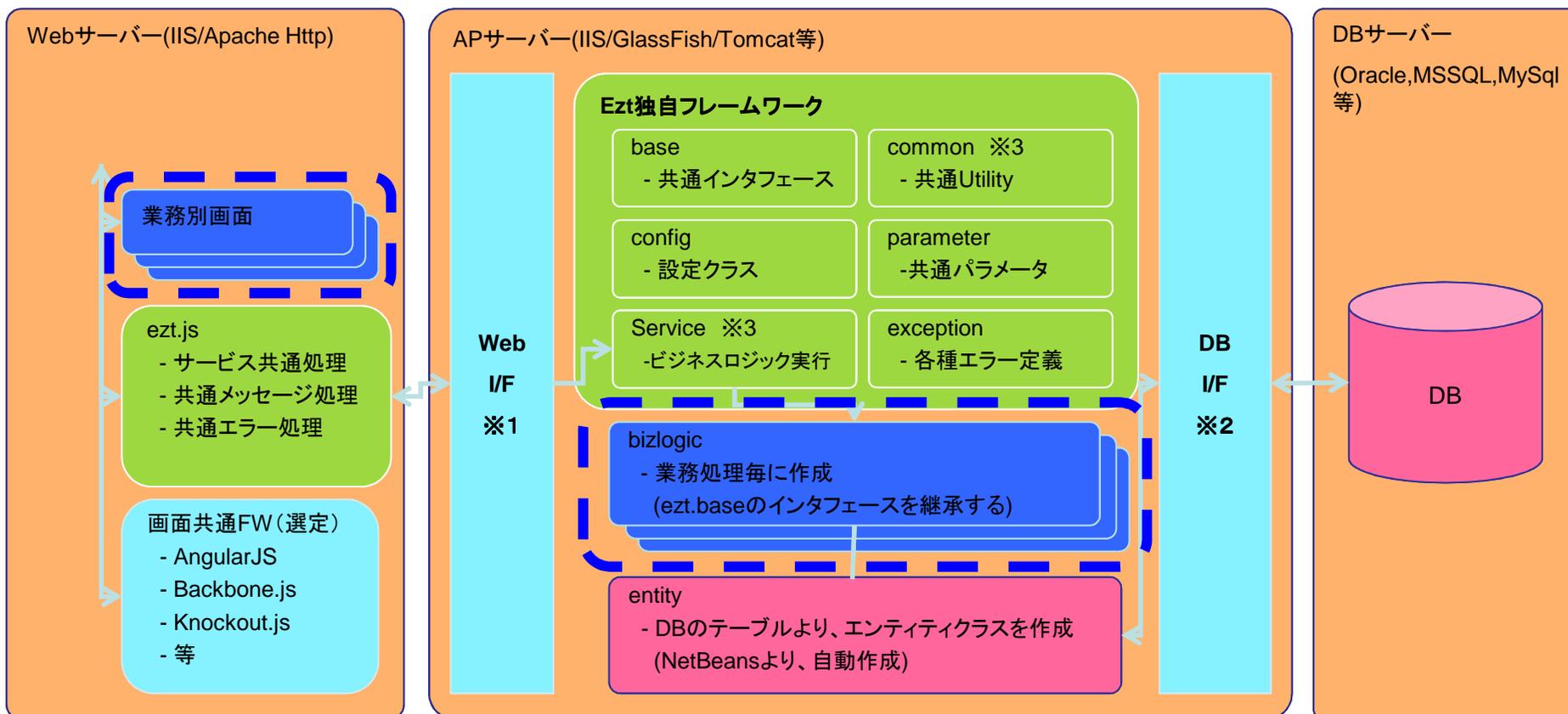
データ通信形式をJson/XMLにすることにより、様々なクライアントに対応可能

### 3. アプリケーションサービス基盤イメージ

お客様の開発範囲  
(サービス基盤はEzt独自FWでカバー)

- : 言語仕様によって変わる部分(例はJavaの場合)
- : Ezt独自フレームワークの共通ロジック
- : DB関連、自動でDAO(Data Access Object)を生成する
- : 業務単位で追加される部分、実際の製造はこの部分だけ実施する

※3 弊社で共通ロジックをカスタマイズする場合の対象



View-Controller

Model

- ※1 Web I/F (Web API)
- .Net : Wcf Service
  - Java : JAX-RS

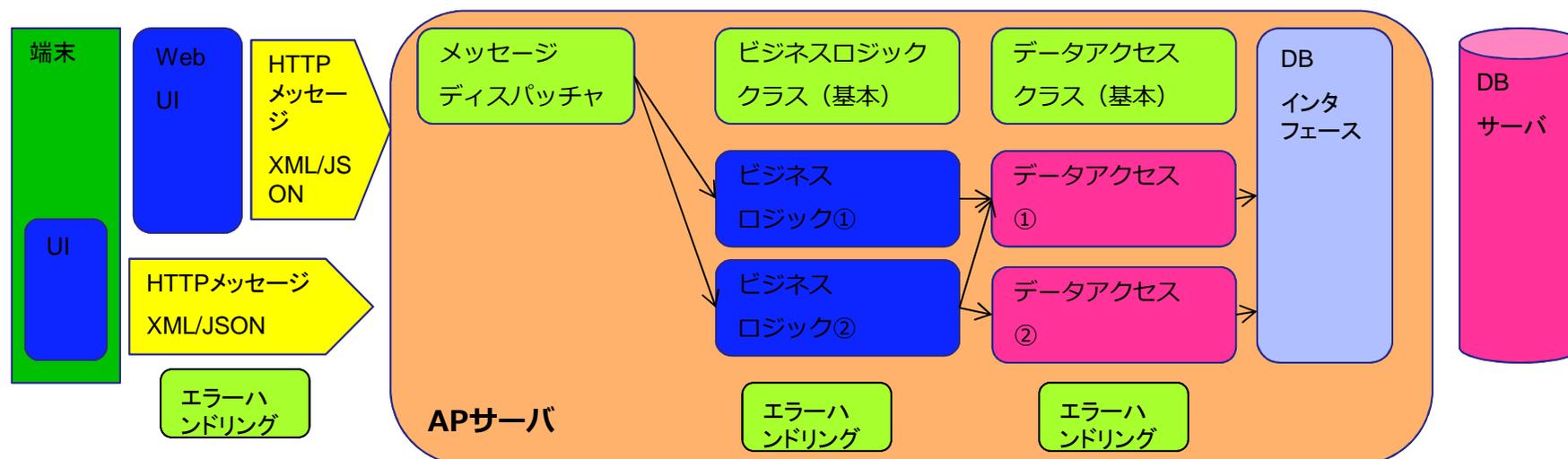
- ※2 DB I/F
- .NET : Entity Framework
  - Java : JPA

## 4. アプリケーション構築イメージ

フレームワークで提供される機能

- ・メッセージ処理のクラス（メッセージディスパッチャ）
- ・エラー処理のクラス（エラーハンドリング）
- ・ビジネスロジックのクラス
- ・データアクセスのクラス
- ・データベースアクセスインタフェース

メッセージの遷移、データアクセスについては、設計時に定義する。UI・ビジネスロジックはユーザ側でも開発が可能とする。



## 【参考資料】開発比較表



		開発期間	共通ロジック開発量	業務ロジック開発量
標準的開発	.NET	大 (標準的)	大	大
	JAVA	大 (.NETに比べ多い)	大	大
EZT独自フレームワークでの開発	.NET JAVA 共通	中 (.NETより短縮可能)	開発は不要  ただし、弊社にてカスタマイズし提供します	中 (共通ルールに基づく為、シンプルに作成)

## 【参考資料】EZT独自フレームワーク言語別対応表



	Java	.Net
ベースバージョン	Java EE 7	.Net Framework 4.5
WepApi (Restful Service)	JAX-RS	Wcf Service
通信データ形式	Json/XML	Json/XML
ビジネスロジック処理	EZT共通ロジック	EZT共通ロジック
DAO (Data Access Object)	JPA (Java Persistence API) JPQL (Java Persistence Query Language)	Entity Framework LINQ to Entities
DB	Oracle, MSSQL, MySql, PostgreSql ...	Oracle, MSSQL

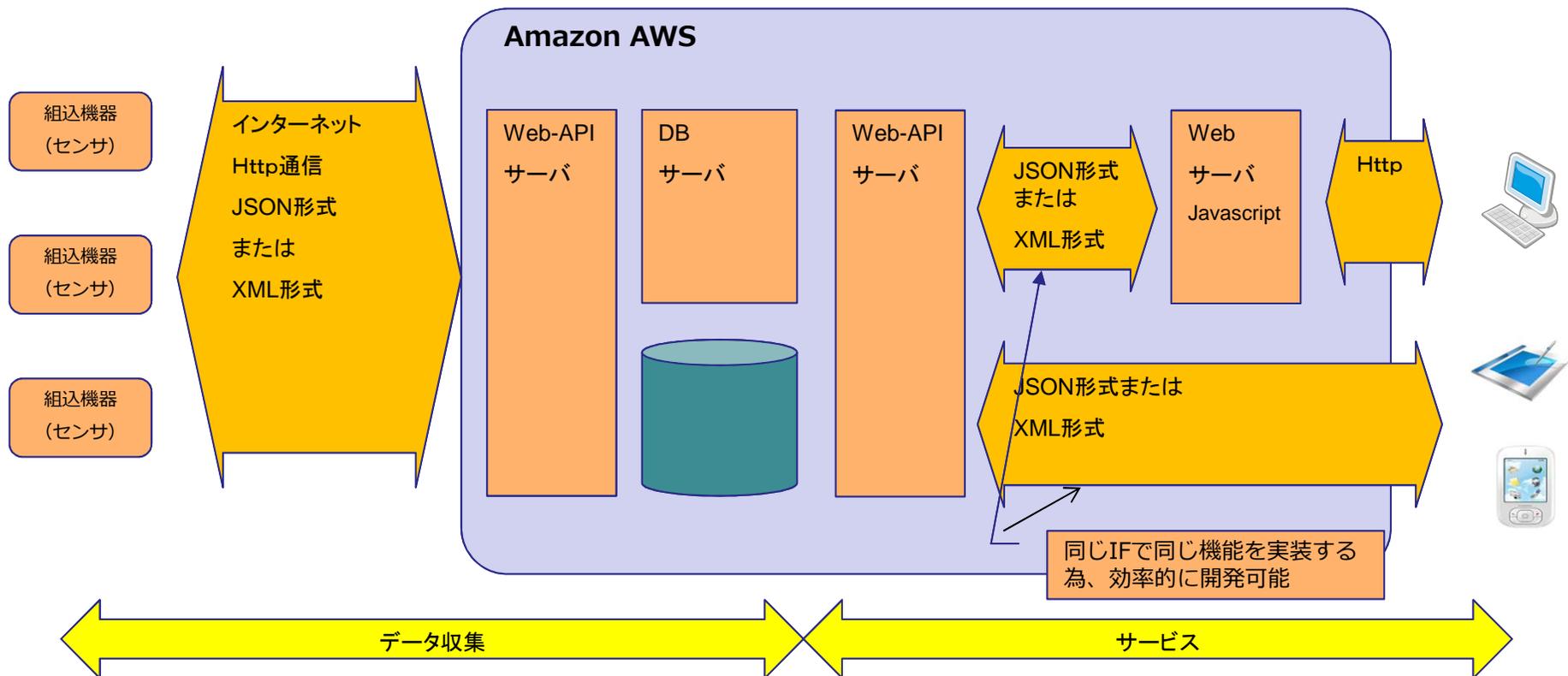


分類	EZT独自フレームワーク
メリット	<ol style="list-style-type: none"> <li>1. ワーカ（メッセージ単位のビジネスロジック）を制御するフレームワークを提供するので、SEが必要機能を定義し、PGがコーディングする作業分担が容易</li> <li>2. デザインパターン（オブジェクト指向における再利用性が高いクラスやライブラリ的设计）を意識しないで開発が可能（通常はオブジェクト指向やMVCモデル、EJBの理解までに時間が掛かる）</li> <li>3. コントロール部がシンプルなため軽量かつ軽快に動作する。</li> <li>4. パッケージに縛られないため、オープンソースの組合せが自由。（Linux+Postgresqlなど）</li> <li>5. 共通部分をコードではなく、仕組みとして上位クラス化することで生成コードの量を減らせる</li> <li>6. 現場に合わせた要望への柔軟な対応が可能</li> <li>7. I/F部が軽量でシンプルな構造になっている為、Java EEのバージョンアップ等にも共通部のみ新しいバージョンのAPIとして管理が出来る</li> <li>8. 最新バージョンの技術を利用するため、技術的な処理の寿命が長い</li> </ol>

# 【参考資料】利用例



センサー等の機器からデータを収集し、収集したデータを元にサービスを行う仕組みに利用する場合の例  
サービスインフラはAmazonAWSを利用することにより初期費用なしに安価に開始できます。



WebAPIサーバ（アプリケーションサーバ）ではPCでもスマホ等のモバイル端末でも、同じ機能を実装し、シンプルに開発が可能  
PC、モバイル端末では、端末の特性に合ったUIを開発することが可能になります。

# 【参考資料】IoT時代に向けた情報収集フレームワーク

IoTという3文字が昨年よりよく聞かれるようになりました。ビッグデータとともにIT業界では最近のキーワードとなっています。

IoTでは、世の中に存在する物が、すべてネットワークで接続され、様々な情報を収集し、サービスとして提供できる環境の構築が可能になってきました。SFの世界に出てきそうな、すべての情報がビッグデータに収集・分析され、必要な情報として送り出されていく世界ができつつあります。

弊社では、これまで、培った技術で、携帯電話、スマートフォン、スマートウォッチ、スマートバンド、センサ機器等の様々な端末から、情報の収集・蓄積・サービス提供を実現するフレームワークを構築し提供いたします。

