intra-mart®
intra-mart accelerates your business.

intra-mart Accel Platform
- 製品カスタマイズ方法に関して -

株式会社ＮＴＴデータイントラマート
開発本部
2015/4/2

intra-mart
intra-mart accelerates your business.

はじめに

iAPリリース後、多くの事例が増えてまいりました。その中でもiAP製品自身に直接カスタマイズをしているケースが多く見られます。

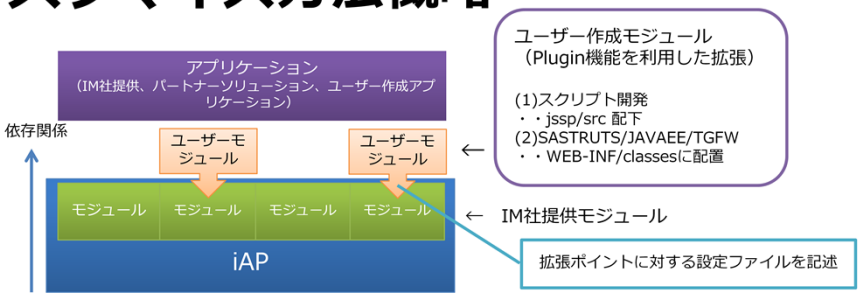
しかし、製品に対して安易に直接カスタマイズしてしまうことは、あとからのiAPのアップデートに対する柔軟性を欠くこととなり、結果としてIM社から提供されるアップデート（機能強化、機能追加モジュール）の恩恵を受けることができなくなります。

特にiAP以降は「基盤ごとバージョンアップ」の概念がなくなるため、継続的にリリースされるアップデートを適用しやすい構造を保持しておくことは、今後のIT投資を効率化する上でも、またアプリケーションのメンテナンス費用を削減する上でも重要な事です。

当資料ではiAP製品に対するカスタマイズの推奨方法を解説しています。ぜひこの手法を理解した上で、ソースコードへの安易な直接カスタマイズを避けていただくようお願い致します。

intra-mart
intra-mart accelerates your business.

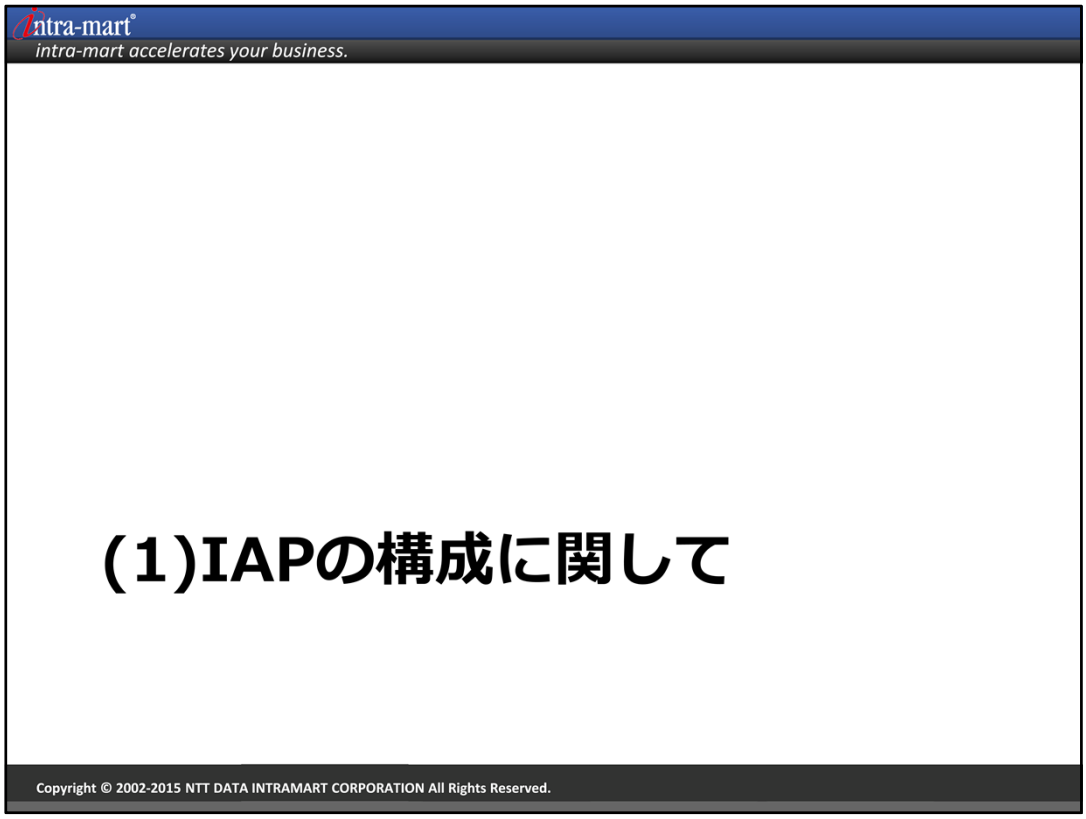
カスタマイズ方法概略



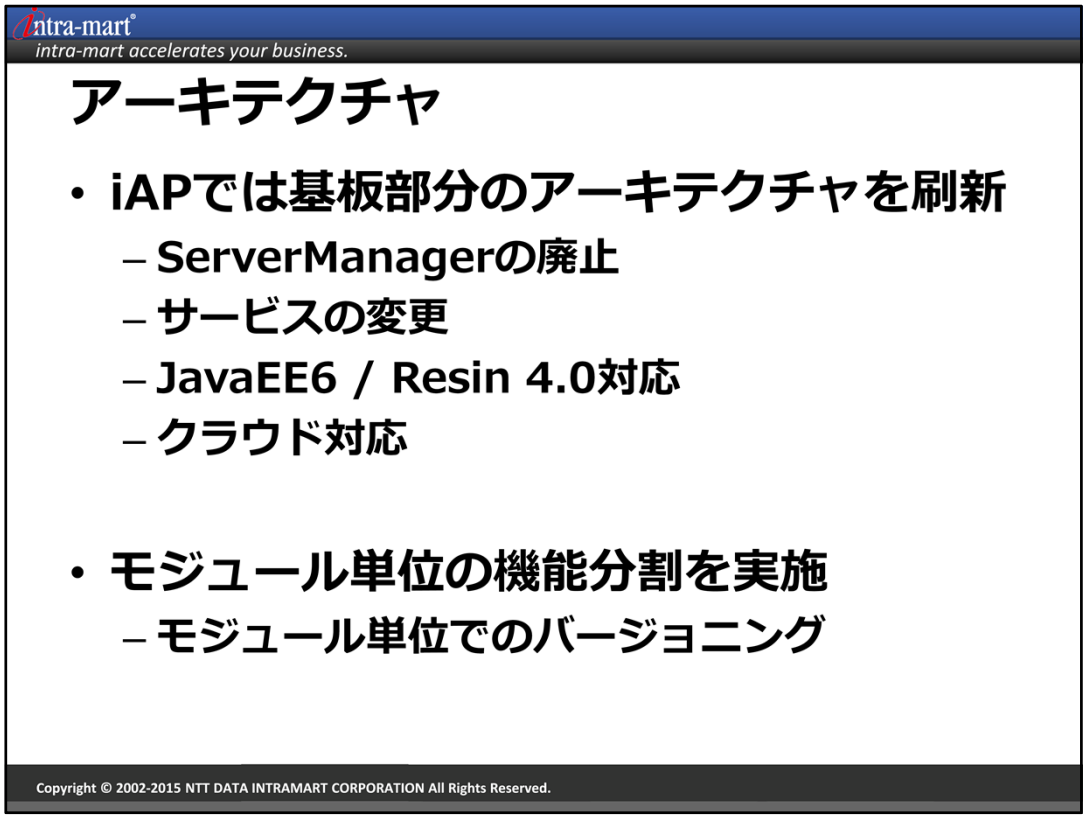
IM社のサポートの考え方（重要）

・iAP以降、アップデートは上図のモジュール単位に実施されます（モジュールに対する個別パッチは緊急時以外、通常リリースされません）。
・このガイドに基づいたカスタマイズ方法であれば、モジュールのアップデート時にもユーザーモジュール/アプリケーションの動作はIM社により保証されます。
（その上で、ユーザーモジュール/アプリケーションの動作確認テストを実施するかどうかは各社判断となります）
・拡張サポートでは、モジュールのアップデート時に影響を受ける可能性のあるユーザーモジュールやアプリケーションについて個別レポートを提出できます。これにより動作確認テストの範囲を絞り込むことができます（詳細はご連絡ください）。

目次



iAPのおさらいとして、構成に関する説明を行います。



iAPでは、アーキテクチャを刷新しました。

これまでiWPに存在したバックエンドサービスの構成を見直し、各サービス機能の改修を実施しております。

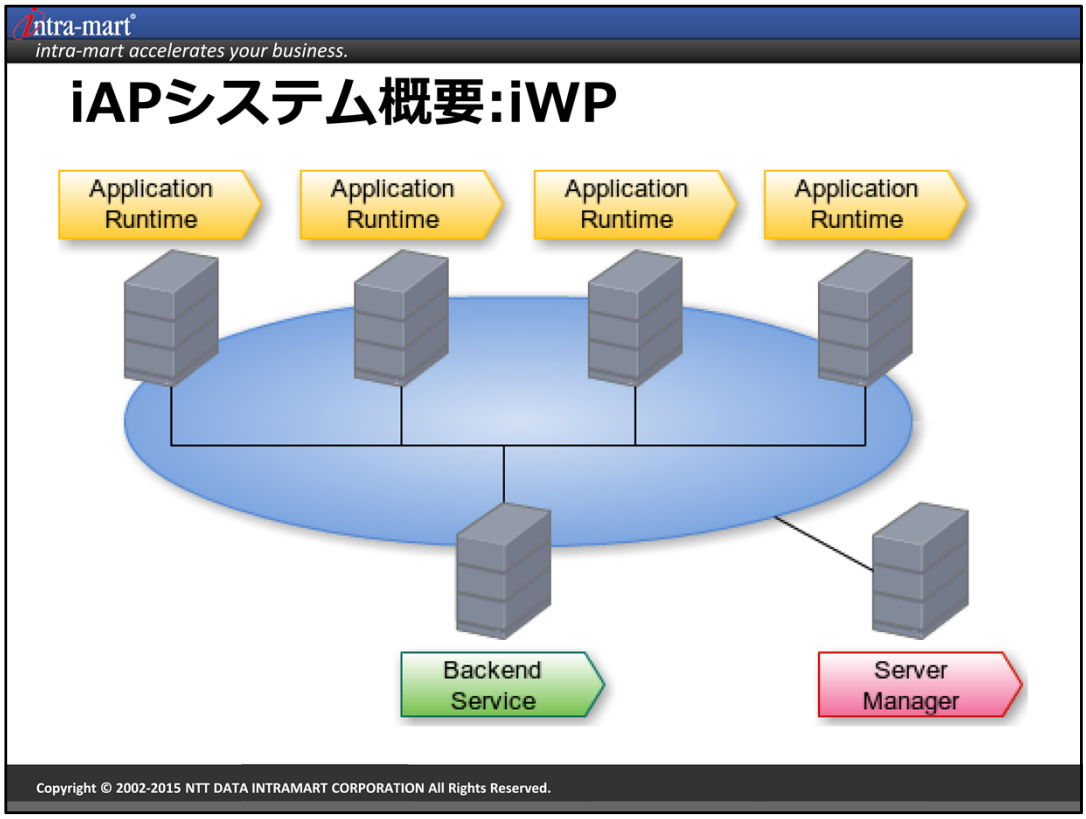
また、これまでバックエンドサービスを個別に用意していた部分を見直し、バックエンドサービスを内包するアーキテクチャに変更しております。

また、JavaEE技術におけるJavaEE6 WebProfile環境で動作するよう改修を実施しました、既に仕様が公開されているJavaEE7、仕様検討中であるJavaEE8等にも追随出来るよう設計されております。

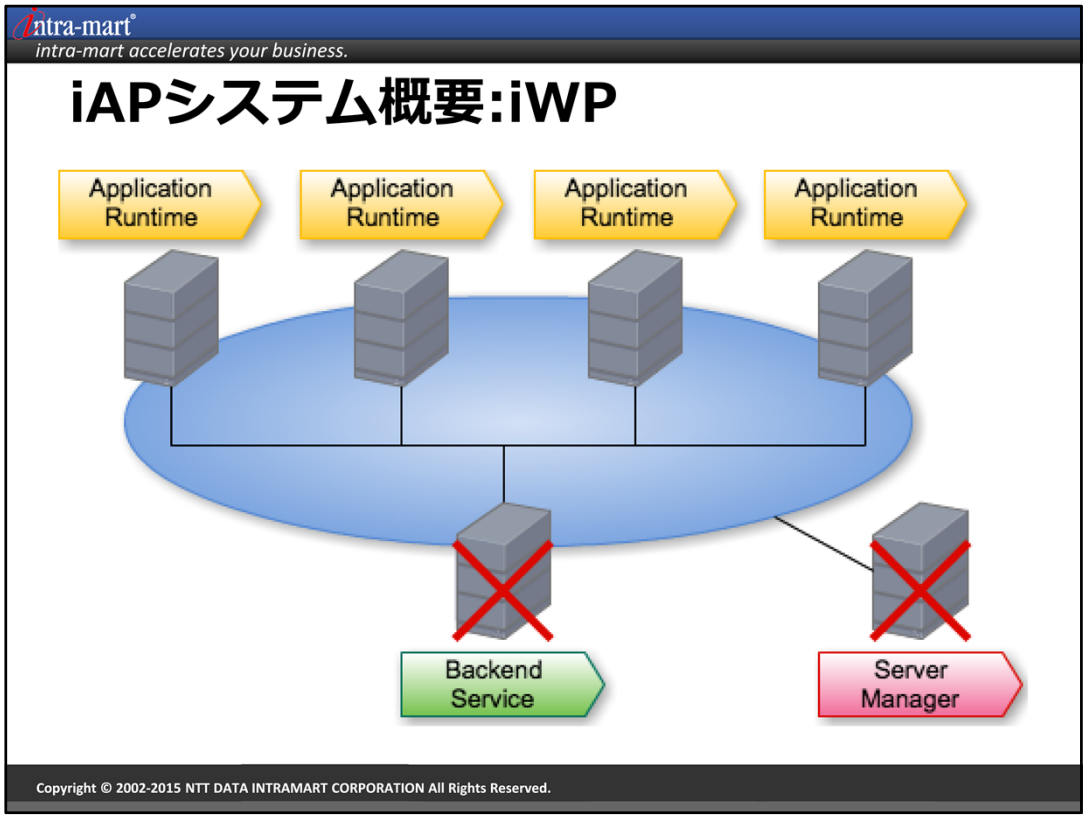
現行では、JavaEE6に対応しているResin, WebSphere, Weblogicに対応しています。

アーキテクチャの刷新とともに、これまでの内部構造を見直し、モジュールという単位での機能分割を実施しております。

これにより、機能単位でのパッチ提供、機能間の連携部分、互換性の向上を行うことが出来ました。



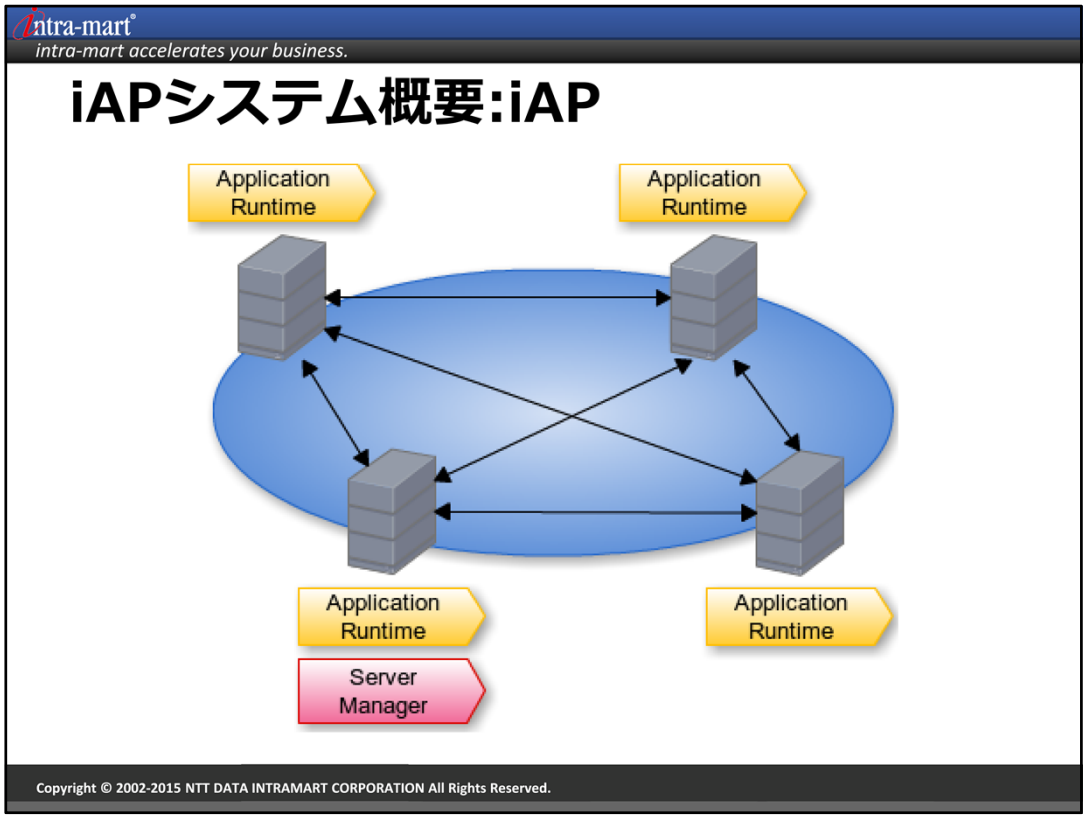
バックエンドサービス部分の変更を詳細に説明します。
これまでのiWPでの構成は図のような構成となっていました。



この構成は、すべてのサービスは必ずServer Managerにより管理されています。

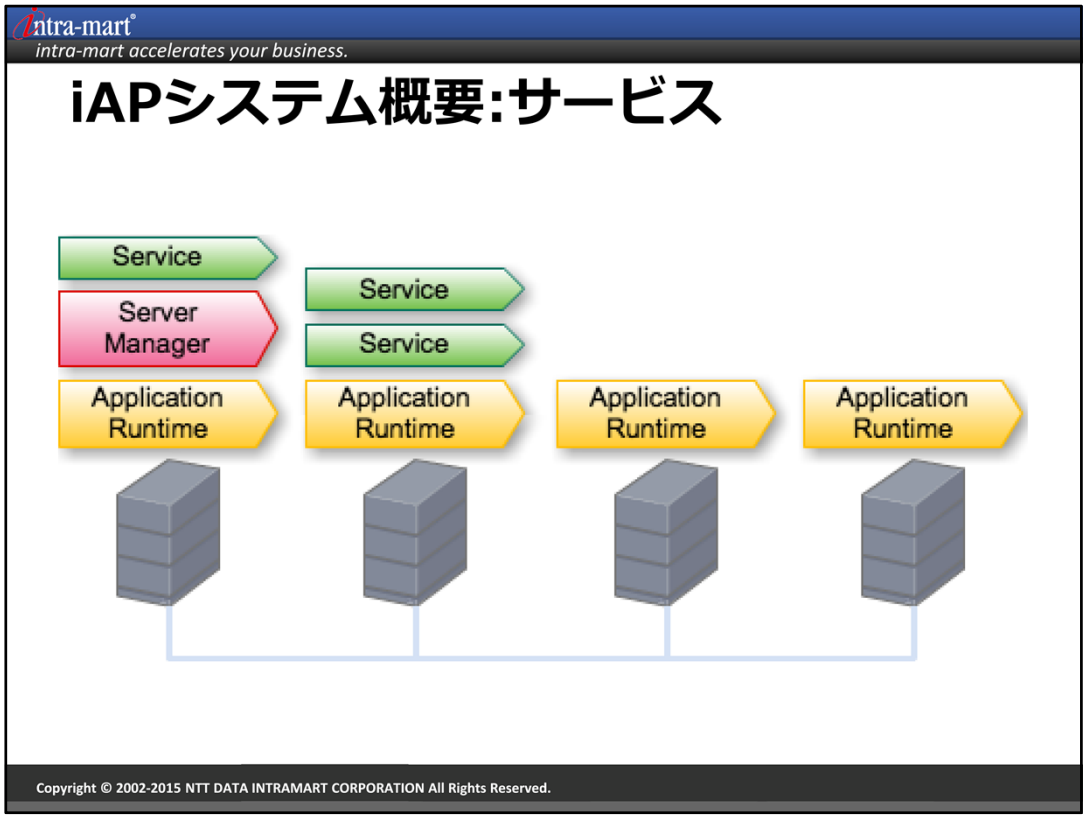
そして、バックエンドサービス及びサーバマネージャが単一障害点となっておりました。

バックエンドサービス自体は、アクティブスタンバイ構成を組むことが可能でしたが、Server Managerに関しては行えませんでした。



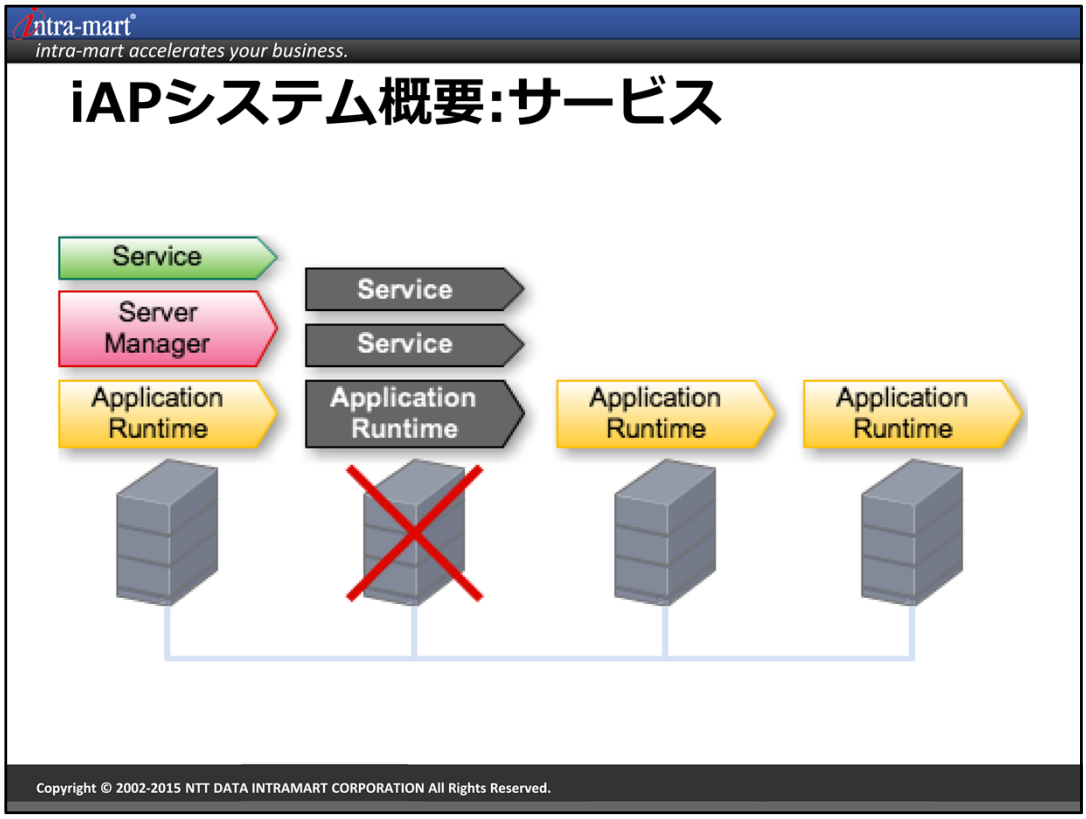
iAPではこれまでの構成を見直し、図のように各機能がお互いに監視し、協調動作を行う仕組みとなりました。

Server Manager及び一部のサービスは残っていますが、iAPの仕組みに内包される形となりiAPを構成するクラスタの中で自動的に割り当てられる仕組みに変更されています。

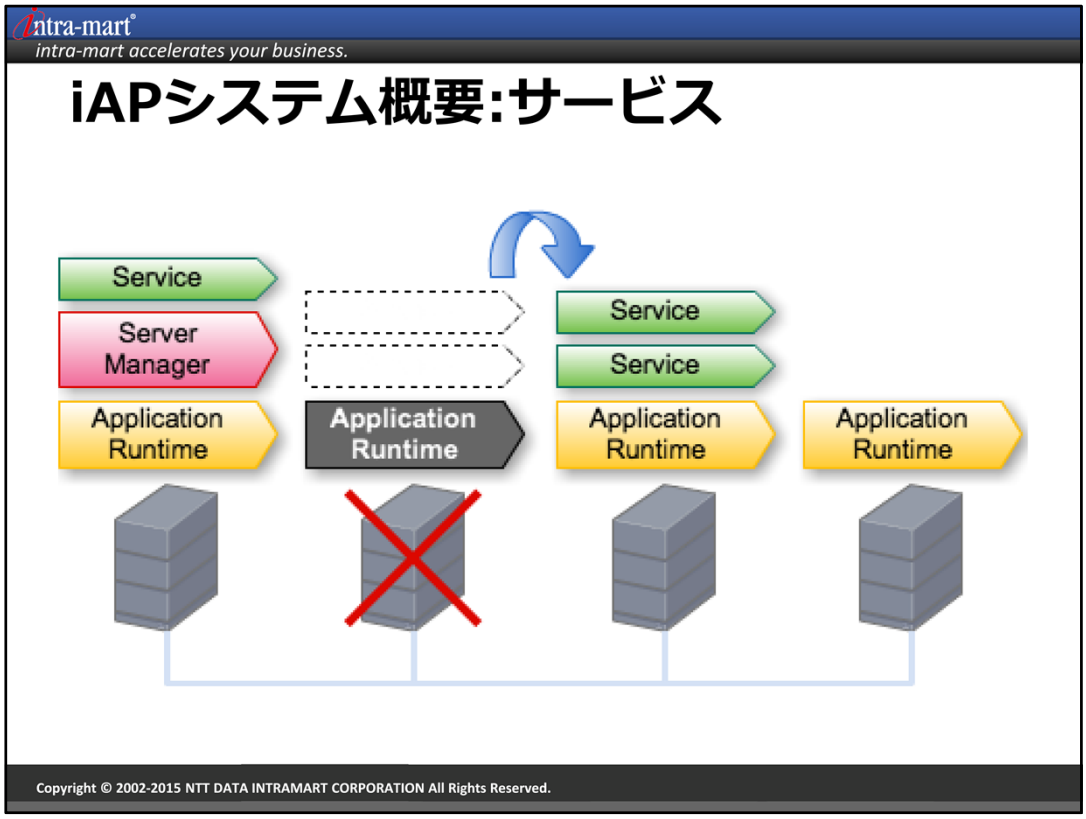


Server Manager及び、各サービスの変更点に関して説明します。
分散環境では、自動的にServer Manager及びServiceが特定のサーバ上で起動します。

Server Manager及びServiceの起動場所に関しては設定により指定する事も可能ですが
デフォルトでは全てのサーバ上で自動的に選出されるようになっております。



この状態において、特定のサーバがダウンした場合、そのサーバ上で動作していたサービスは停止してしまいます。

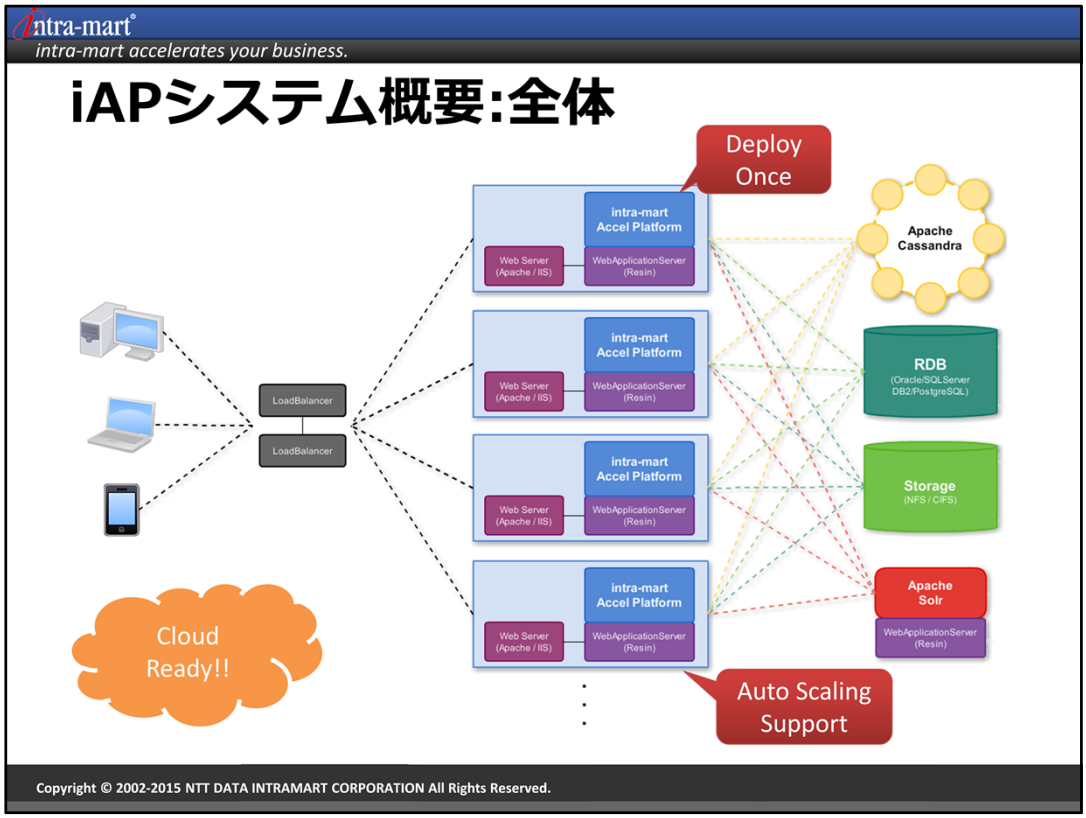


iAPでは、サーバの停止を検出すると、自動的に残る他のサーバ上で停止したサービスを起動し処理を継続します。

これにより、これまで存在した単一障害点が無くなります、またサーバの追加も容易に行えるようになっております。

(サーバ停止の検出及びサービスの切り替えには数秒～10秒程度を要します。)

(図にしますと構成に大幅な変更があるように見えますが、これまでの各サービスの持つ機能等は互換性を保っている状態ですので利用する側からは大きな変更点はございません。)



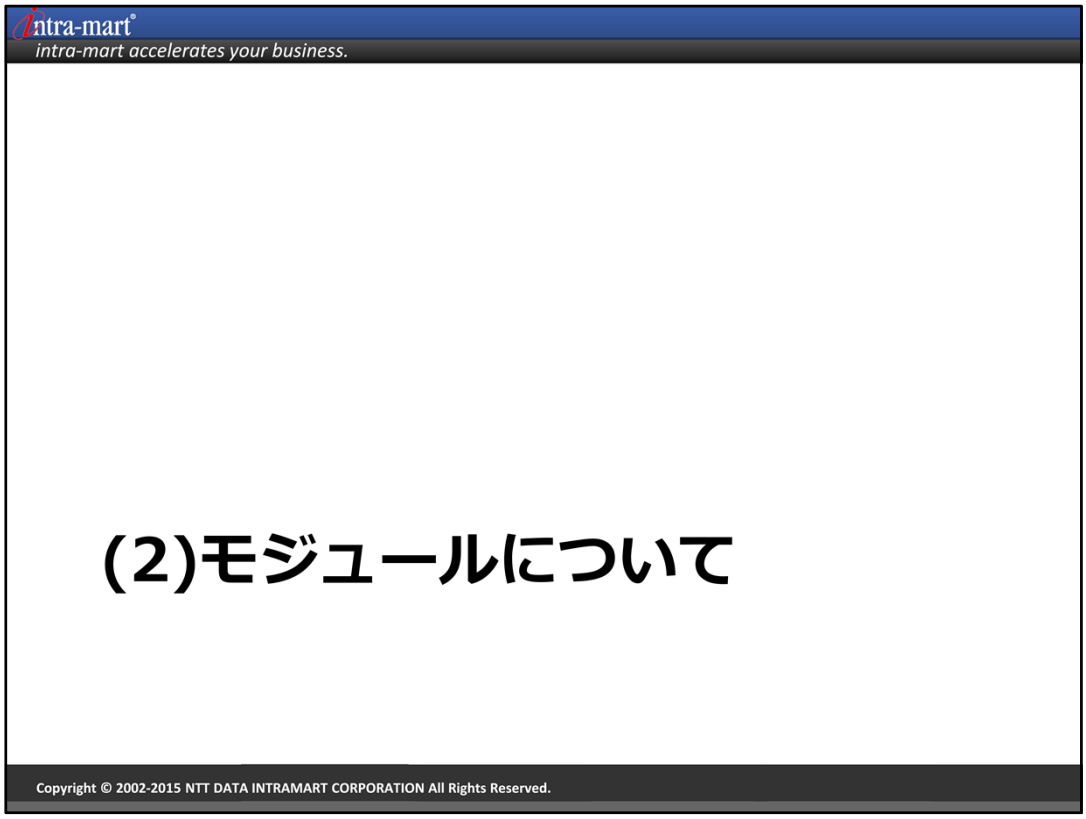
システムの全体図です。

iAPでは今回のアーキテクチャ変更により、これまでよりクラウド環境での運用が行い易くなっております。

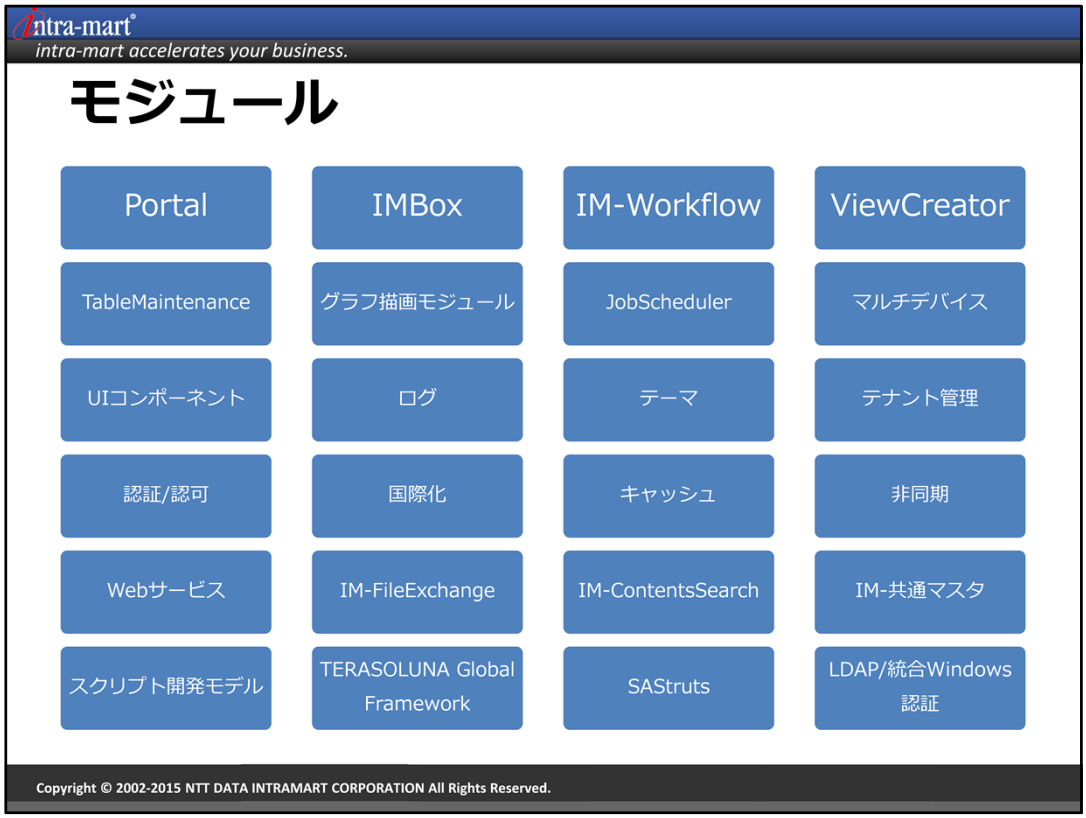
負荷状況に応じた動的なサーバ追加による水平スケールが可能となっています。

また、これまでiWPでは全てのサーバに対してインストール作業が必要でしたが、iAPではwarファイルをデプロイすることにより全てのサーバに対してデプロイされた内容が反映されます。

また、im-Jugglingツールにより任意の機能の追加、パッチの適用等が一括で行えるようになり、これまでそれぞれのサーバにインストールしていた作業が全て不要となっております。



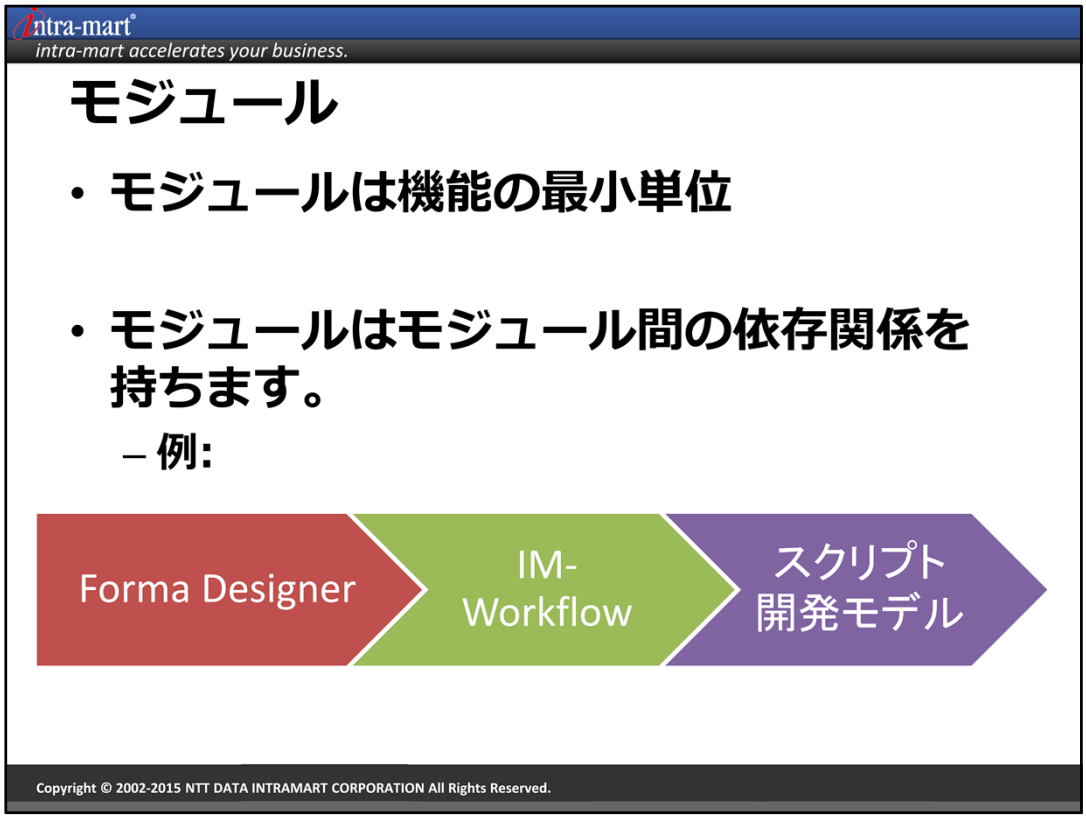
次にモジュールに関して説明します。



iAPでは、図のような粒度で機能を分割し、モジュール化を実施しました。

これまでiWPでは、エディションによる差異はありましたが、全ての機能がインストールされる形式をとっておりました。

iAPではこれらのモジュールを任意に付け外し出来るようになりました。



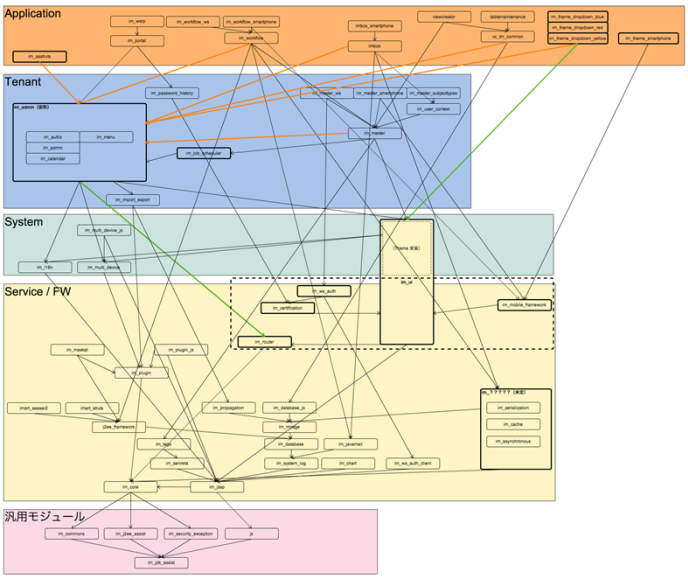
モジュールは依存関係を持ちます。

例えば、Forma Designerはim-Workflowと連携しています、その為Forma Designerはim-Workflowに依存しています。

また、Forma, im-Workflow共にスクリプト開発モデルによる実装が行われている為、それぞれのモジュールはスクリプト開発モデルに依存しています。

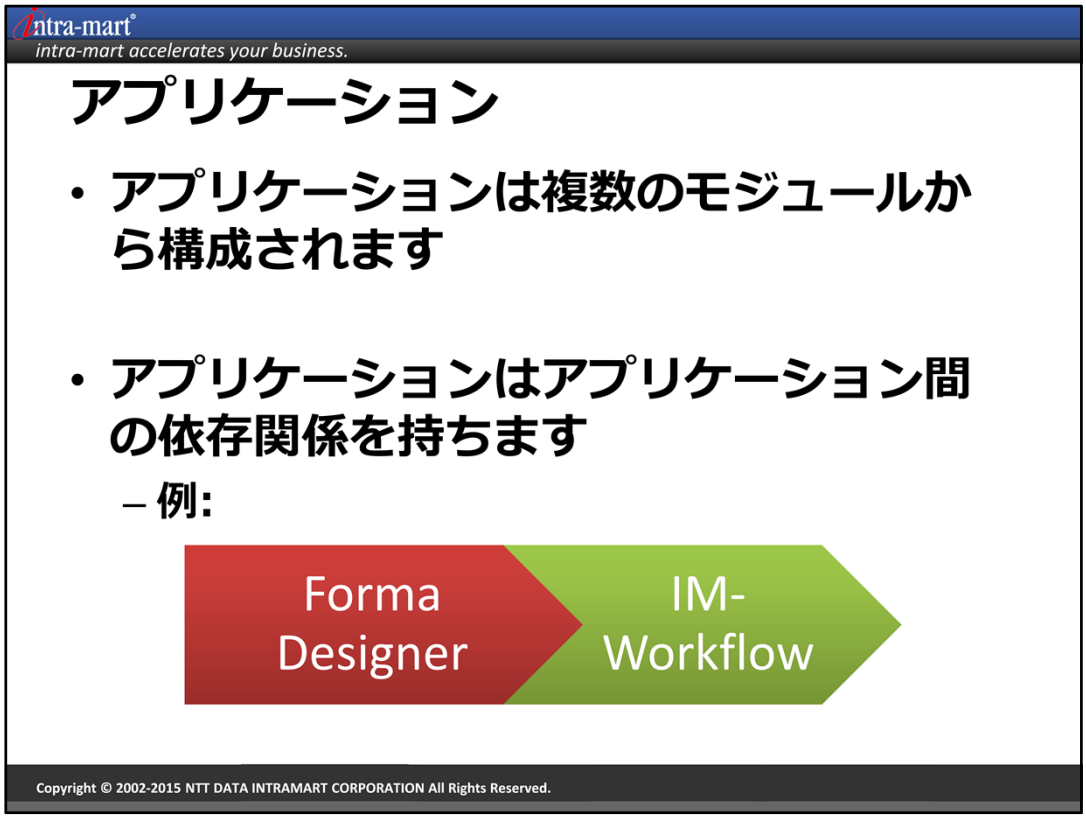
intra-mart
intra-mart accelerates your business.

モジュール:依存関係



内部的なモジュールの依存関係の例です。

このようにモジュール間の依存関係を全て見直し、かつそれぞれのモジュールにバージョンを持たせることにより機能変更、機能追加における影響度の把握が可能となり、また部分的なパッチの適用等が実現できるようになりました。



次にアプリケーションです。

これは一般的にアプリケーション/エクステンションという形の

これは、一般的に複数のモジュールを組み合わせた製品という形をとっています。

例えば、iAC, iAD, BIS, Forma等がそれに該当します。

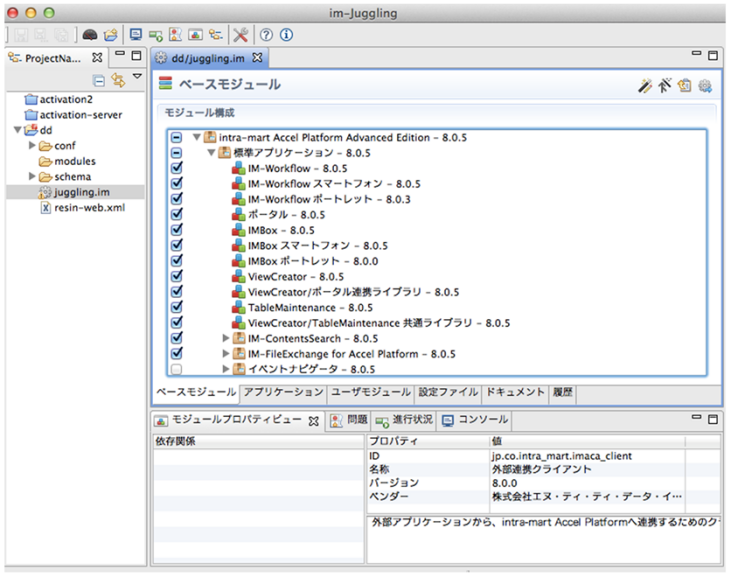
アプリケーションは、それぞれアプリケーション間の依存関係を持ちます。

例えば、BISはFormaを利用していますので、BISとFormaの間には依存関係が存在します。

また、それぞれのアプリケーションのバージョンによる組み合わせも全て管理しています。

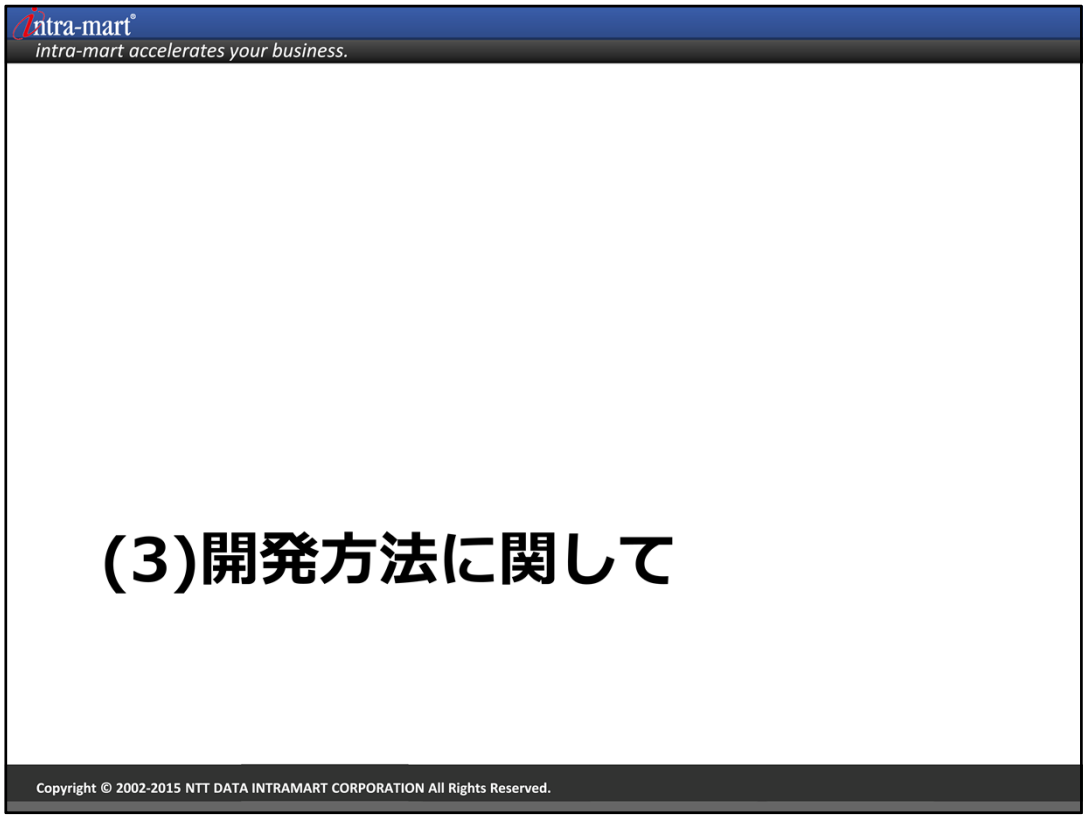
intra-mart
intra-mart accelerates your business.

im-Juggling

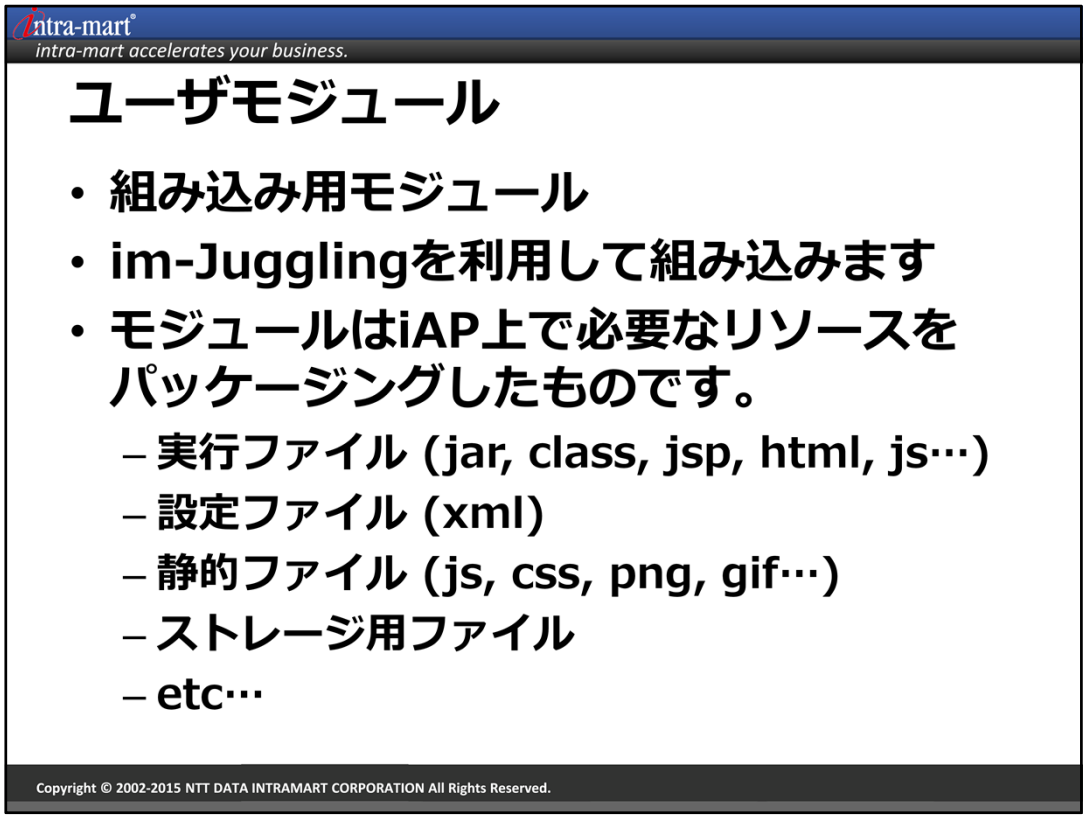


これらのモジュール、アプリケーションは全てim-Jugglingツールにより管理されます。

このim-Jugglingツールは弊社が管理するリポジトリからアプリケーション、モジュールの配信を受けとります。



次に、iAPでの開発についてご紹介致します。



iAP上で何か動作するアプリケーションを作成される場合には、ユーザモジュールという形で独自のモジュールを作成して頂く方法を推奨しております。

ユーザモジュールは、iAP上で動作するアプリケーションに関する資材、例えばhtml, javascript, css等を纏めてパッケージングしたものとなっております。

このユーザモジュールをim-Jugglingを利用して組み込んでいただくことを推奨しております。

ユーザモジュールを利用するメリットは以下の通りです。

可搬性の向上

モジュール化することにより、必要な資材の持ち運び、やりとりが用意になります。

例えば、プロジェクトを横断して利用される汎用的な機能をお持ちの場合等が考えられます。

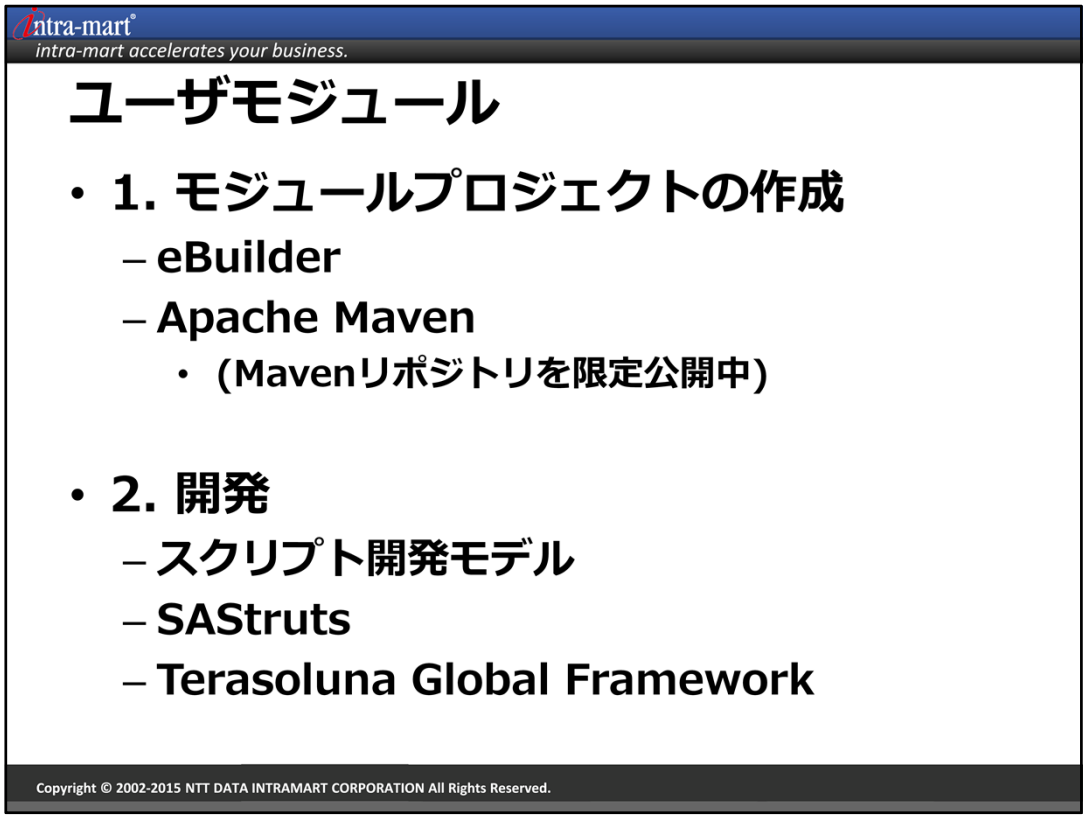
次に依存関係の設定

ユーザモジュールは、どのモジュールに依存しているか、どのバージョンで動作するか等の依存関係の設定を埋め込むことが可能です。

これにより、動作が保証されている環境上にのみ組み込むことが出来るようになります。

最後に環境構築面に関して、iAPでは、WebサーバとしてIIS, Apacheをサポートしています。

これらのWebサーバには通常静的なファイルを配置し、APサーバは動的なリクエストのみ捌くような構成を組むことが可能ですが、im-Juggling, ユーザモジュールを利用することにより、静的ファイルのみの抽出であったり、スタンドアロン構成の為に全てのファイルを抽出する事が可能となります。



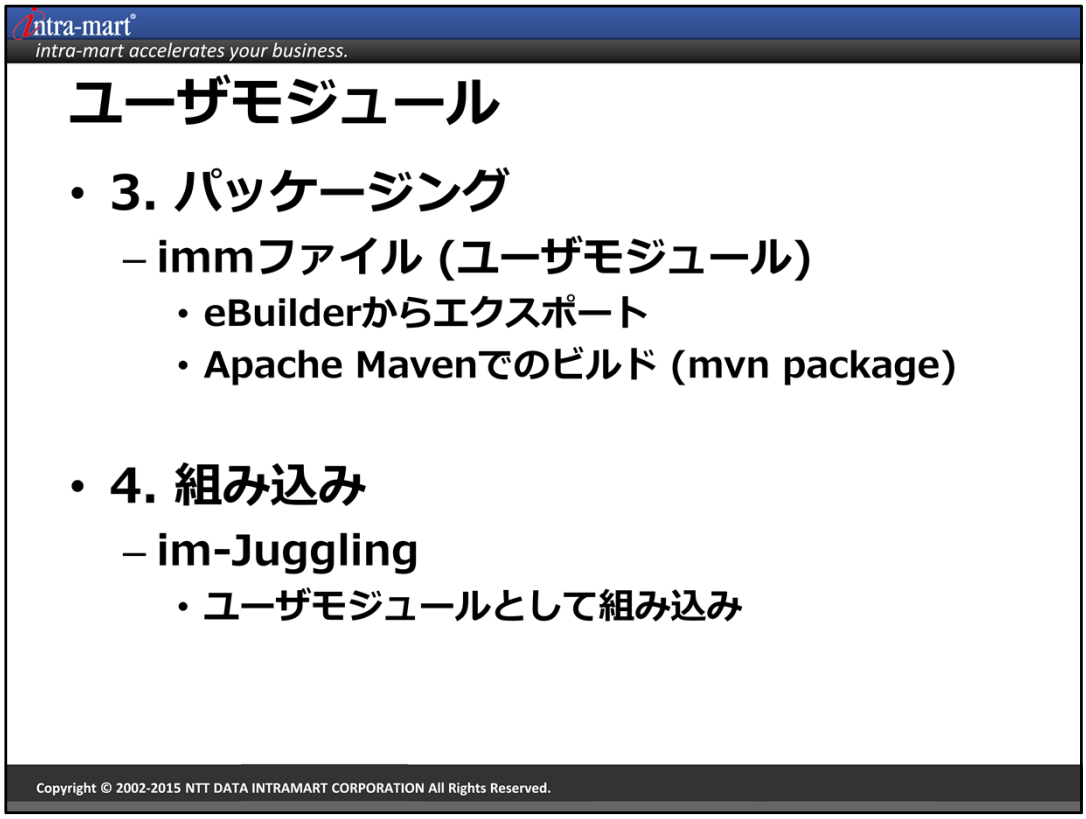
次にユーザモジュールを作成していく流れに関して説明します。

一般的にユーザモジュールを作成する場合、モジュールプロジェクトを作成します。

このモジュールプロジェクトは、通常eBuilderで作成されるかと思いますが、一部限定公開という形でApache Mavenを利用した開発をサポートするためのMavenリポジトリを公開しております。

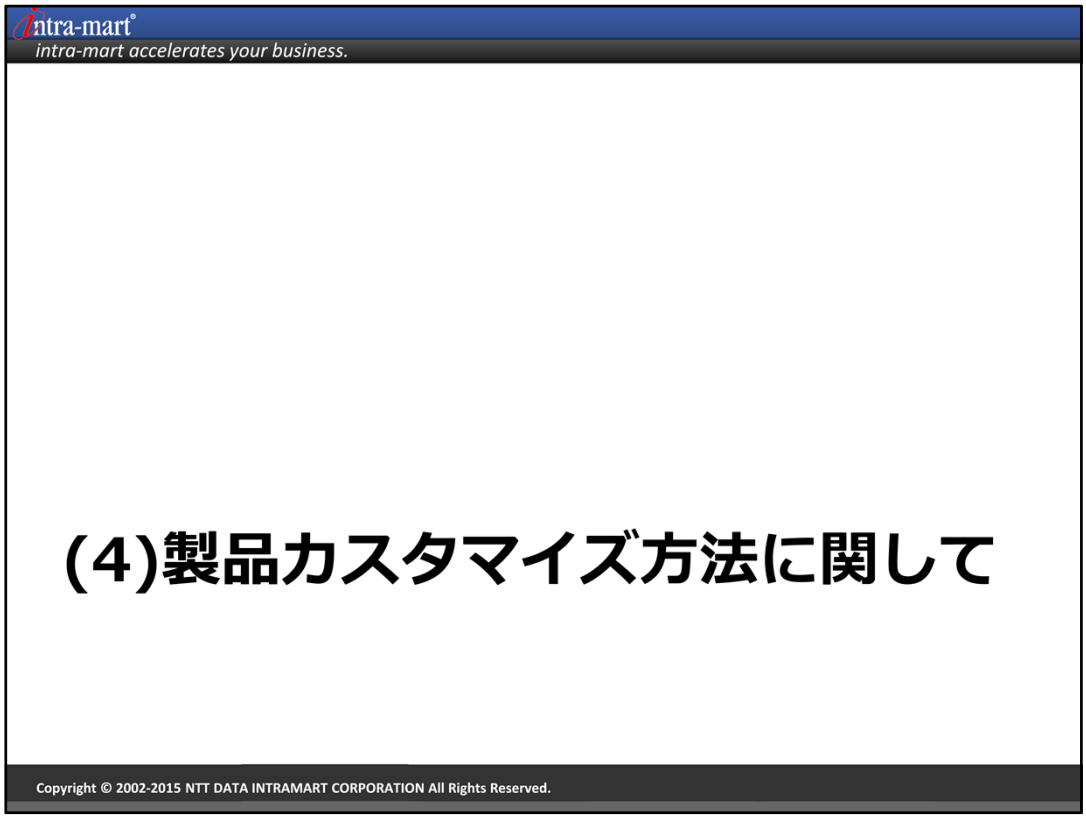
モジュールプロジェクトを作成後、弊社が推奨しております3つの開発モデル、スクリプト開発、SA, Terasoluna等を利用して頂きながら開発を進めて頂きます。

開発部分に関する詳細に関しては割愛させて頂きます。



次に、作成した成果物をパッケージングします。
これにより、ユーザモジュールファイルが作成されます。

作成されたユーザモジュールをim-Jugglingを利用して組み込んで頂く流れとなります。
Im-Jugglingを利用してユーザモジュールを組み込む形となりますが、弊社、開発者ブログにてAntスクリプトによるwarファイルの出力も紹介しておりますのでご覧ください。



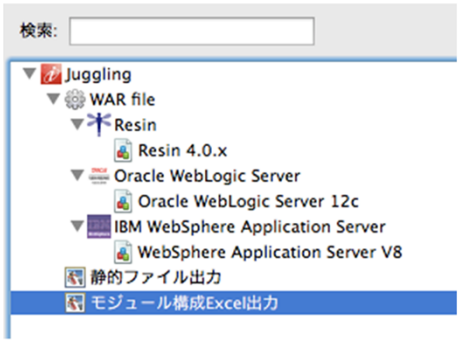
次に製品に関するカスタマイズに関して照会致します。

製品のソースコードは、製品情報一覧よりダウンロード可能となっております。

intra-mart
intra-mart accelerates your business.

カスタマイズ対象のモジュール

- モジュール内に含まれるファイルの確認
  - im-Juggling -> モジュール構成Excel出力



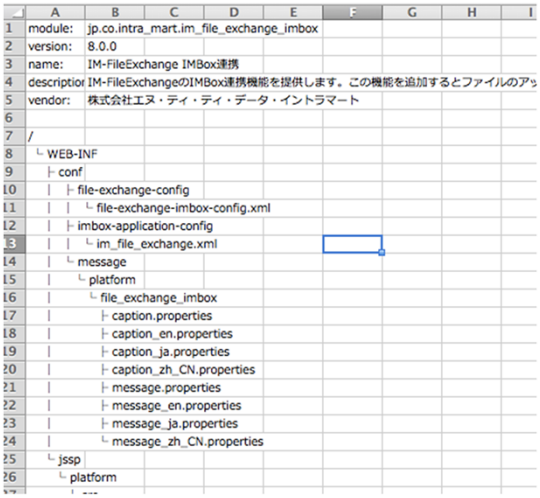
まず、カスタマイズに辺り、im-Jugglingの機能を紹介させて頂きます。

カスタマイズを行う際に、iAPでは全て機能がモジュールという単位に分かれておりますので、カスタマイズ対象のモジュールがどれかを探す場合があるかと思います。

Im-Jugglingには、warファイルを出力するウィザードにて、設定されているモジュール構成をExcelファイルで出力する機能があります。

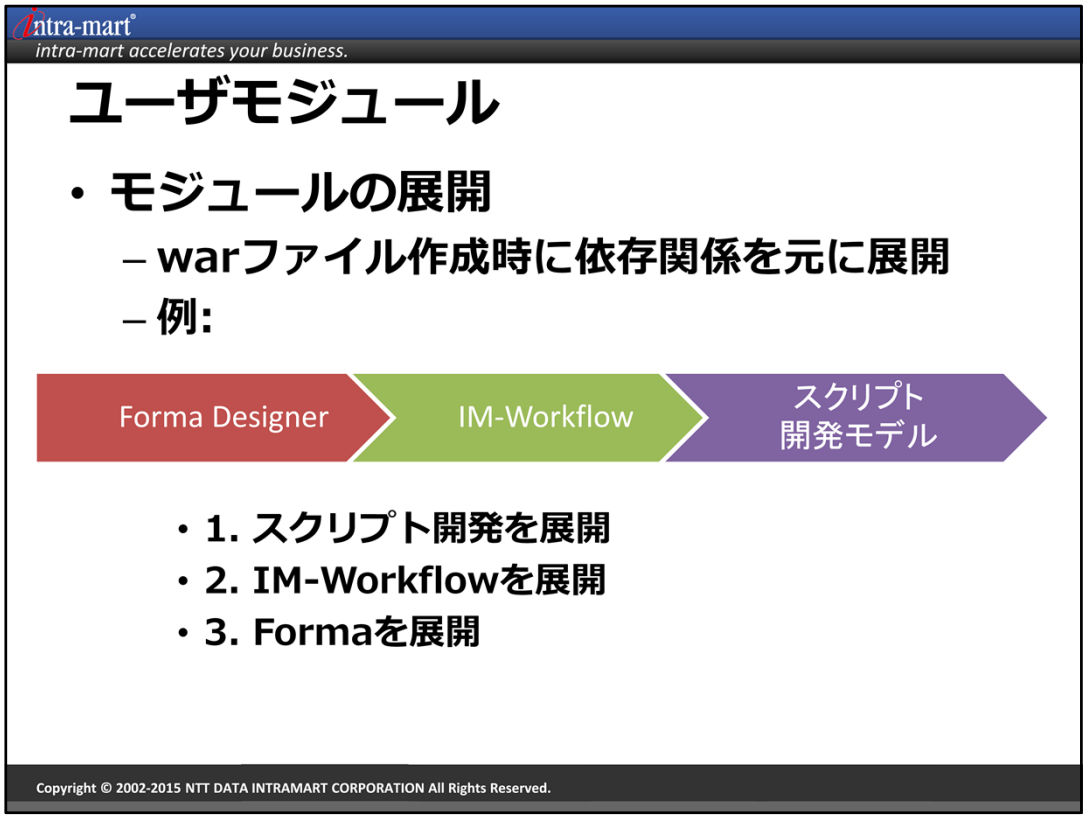
intra-mart
intra-mart accelerates your business.

カスタマイズ対象のモジュール



	A	B	C	D	E	F	G	H	I
1	module:	jp.co.intra_mart.im_file_exchange_imbox							
2	version:	8.0.0							
3	name:	IM-FileExchange IMBox連携							
4	description	IM-FileExchangeのIMBox連携機能を提供します。この機能を追加するとファイルのア							
5	vendor:	株式会社エヌ・ティ・ティ・データ・イントラマート							
6									
7	/								
8	└ WEB-INF								
9	├ conf								
10	│ ├ file-exchange-config								
11	│ │ └ file-exchange-imbox-config.xml								
12	│ ├ imbox-application-config								
13	│ │ └ im_file_exchange.xml								
14	│ └ message								
15	│ └ platform								
16	│ └ file_exchange_imbox								
17	│ ├ caption.properties								
18	│ ├ caption_en.properties								
19	│ ├ caption_ja.properties								
20	│ ├ caption_zh_CN.properties								
21	│ ├ message.properties								
22	│ ├ message_en.properties								
23	│ ├ message_ja.properties								
24	│ └ message_zh_CN.properties								
25	└ jssp								
26	└ platform								

Excelファイルにモジュール構成を出力しますと、モジュール毎にどのようなファイルが含まれているか一覧が出てきますのでこちらからカスタマイズ対象のモジュール、ソースを探していただくことが可能です。



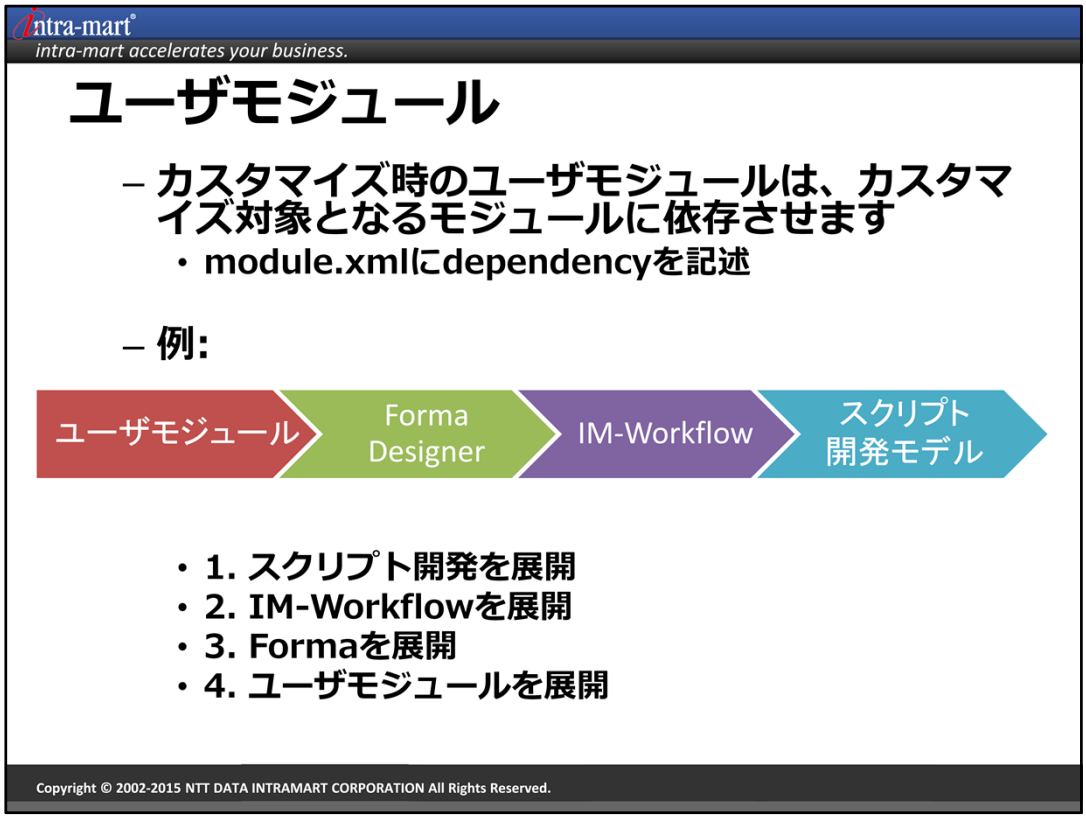
次にim-Jugglingからwarファイルを出力する際の流れを説明します。

im-Jugglingからwarファイルを出力する場合、構成されるモジュールの依存関係を元に、モジュールの展開順序が決定されます。

この例では、Formaはim-Workflowに依存し、im-Workflowはスクリプト開発に依存しているという例ですが、モジュールの展開順序は、

依存されているモジュールから先に展開されます。

この場合は、スクリプト開発が先に展開され、その後im-Workflow, Formaの順に展開されます。



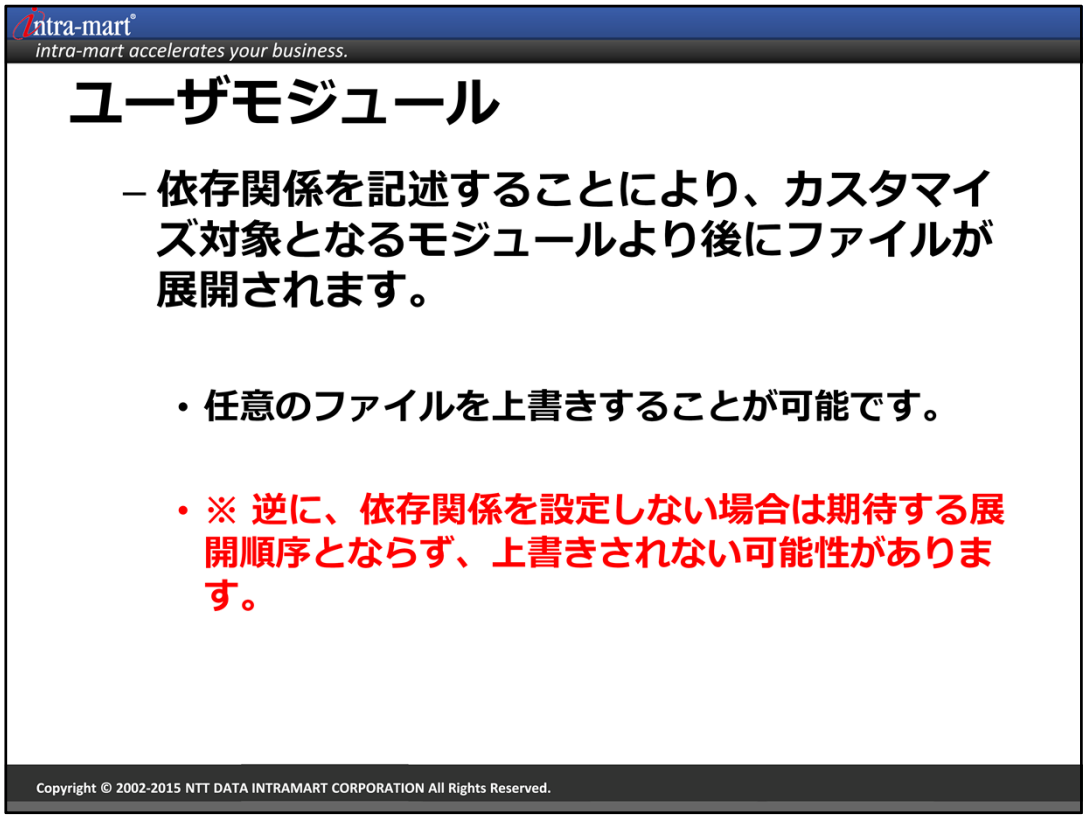
ユーザモジュールを作成される場合、そのユーザモジュールの依存関係を設定することが可能です。

これは、製品のカスタマイズを行ったユーザモジュールを作成する場合、カスタマイズ対象となるモジュールに対して依存関係を記述することにより

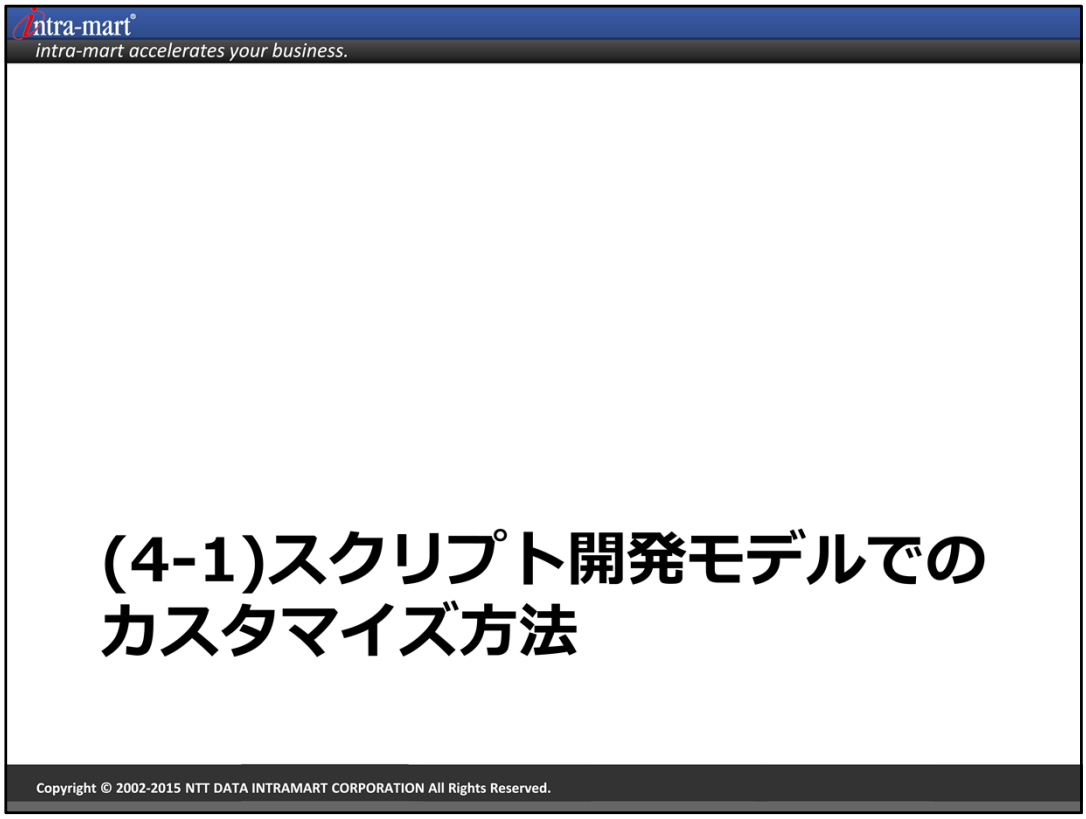
必ずカスタマイズ対象のモジュールの後にユーザモジュールが展開されるようになります。

この例では、先ほどの依存関係にと共に、ユーザモジュールとしてFormaに依存しているユーザモジュールを追加した例です。

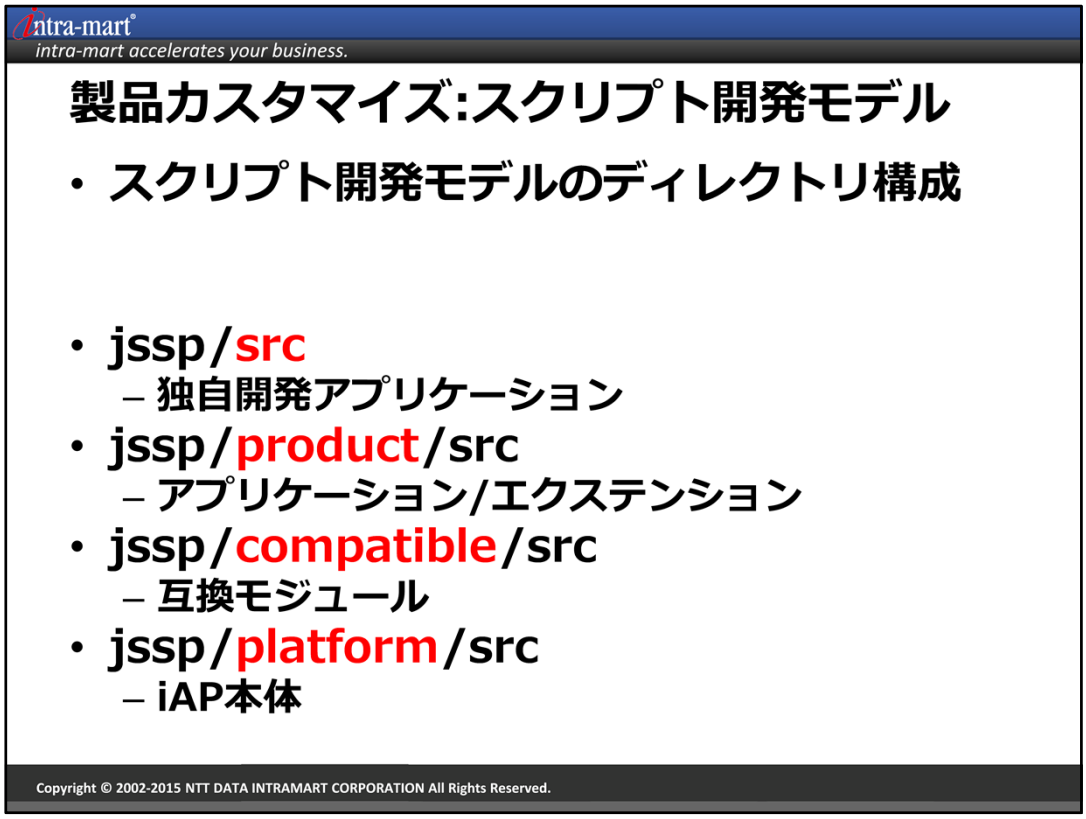
ユーザモジュールはFormaに依存しているため、Formaが展開後にユーザモジュールが展開されるようになります。



カスタマイズの際に注意して頂きたいのは、この依存関係を記述していなかった場合
展開順序がカスタマイズ対象よりも前になってしまう場合がありますのでご注意下さい。



次に、開発モデル毎のカスタマイズ方法について紹介します。



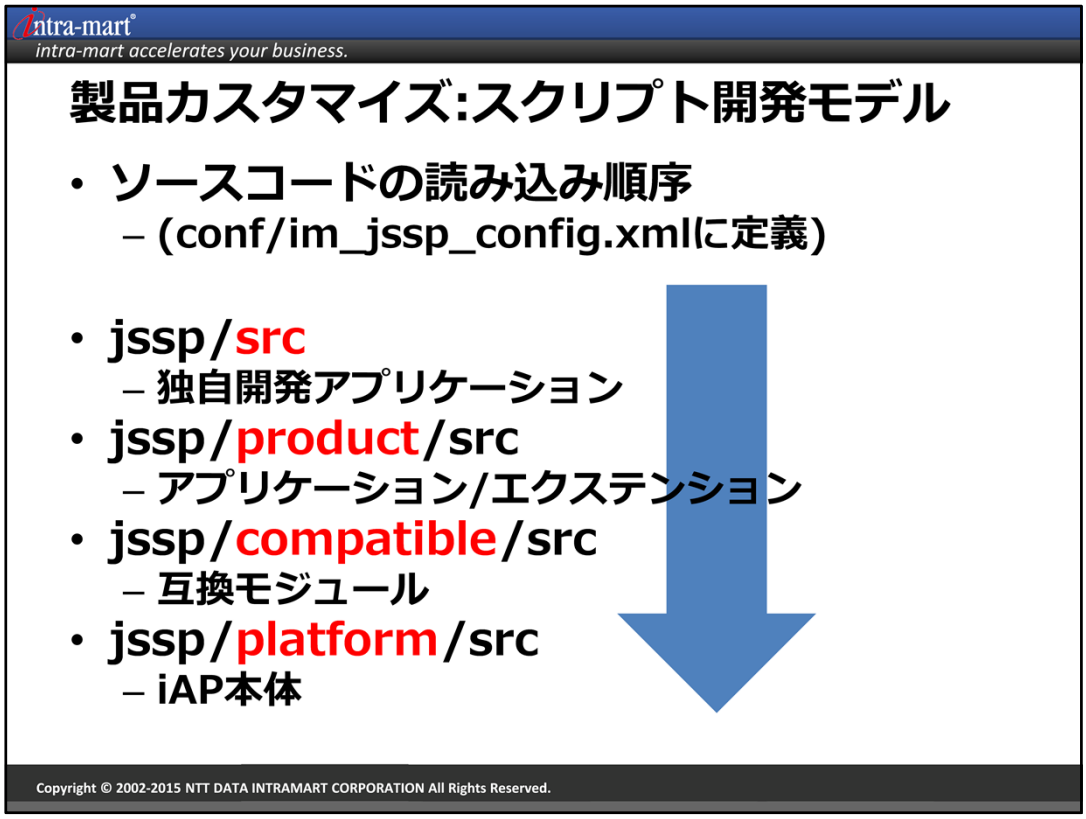
スクリプト開発モデルでは、ソースコードを配置するディレクトが標準で4つ存在します。

Src: 独自開発のアプリケーションを配置するディレクトリです、このディレクトリはインタプリタで動作します。

Product: アプリケーション/エクステンションに関するソースコードが配置されるディレクトリです。

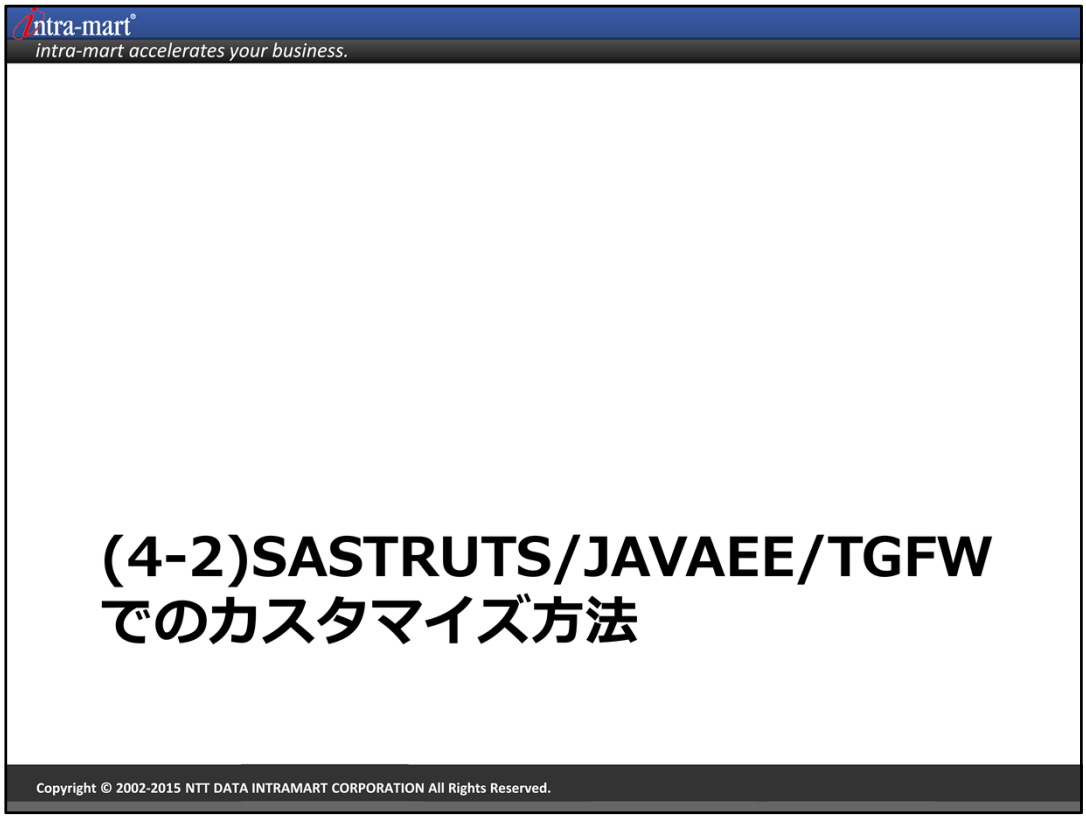
Compatible: 互換モジュールのソースコードが配置されるディレクトリです。

Platform: iAP本体のソースコードが配置されるディレクトリです。



これらのディレクトリは、スクリプト開発モデル実行時に順番にソースコードの検索が行われます。

その為、iAP本体のソースコードをカスタマイズする場合、直接そのファイルをカスタマイズする必要はなく、カスタマイズ対象のソースコードをsrcディレクトリ配下にコピーし、そのファイルを修正することにより本体に変更を加えること無くカスタマイズを行うことが出来ます。



次に、Java言語により実装されているフレームワークを利用したカスタマイズ方法を紹介します。

intra-mart
intra-mart accelerates your business.

拡張機能の利用

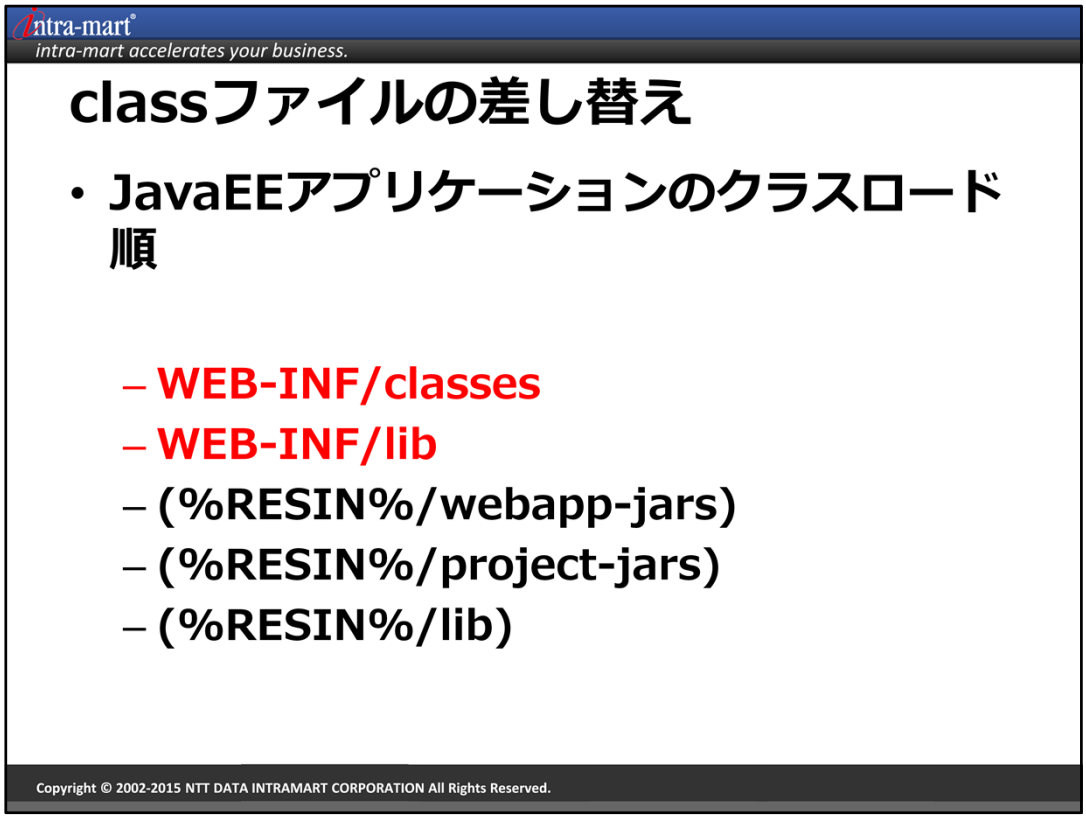
- PluginManager
  - 拡張ポイントを利用した拡張方法
    - 拡張ポイントは各仕様書を参照

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
<extension point="jp.co.intra_mart.foundation.ui.theme.utility.item">
  <item
    id="to_sp"
    name="to_sp"
    classname="jp.co.intra_mart.foundation.ui.theme.ChangeToSPItemProvider"
    version="1.0"
    rank="99"
    target="loginlogout"
    before="true"
    enable="true"
    />
</extension>
</plugin>
```


まずはじめに、これはスクリプト開発モデルも含まれますが、intra-martではPlugin機能を利用した拡張が行えるよう拡張ポイントが存在します。

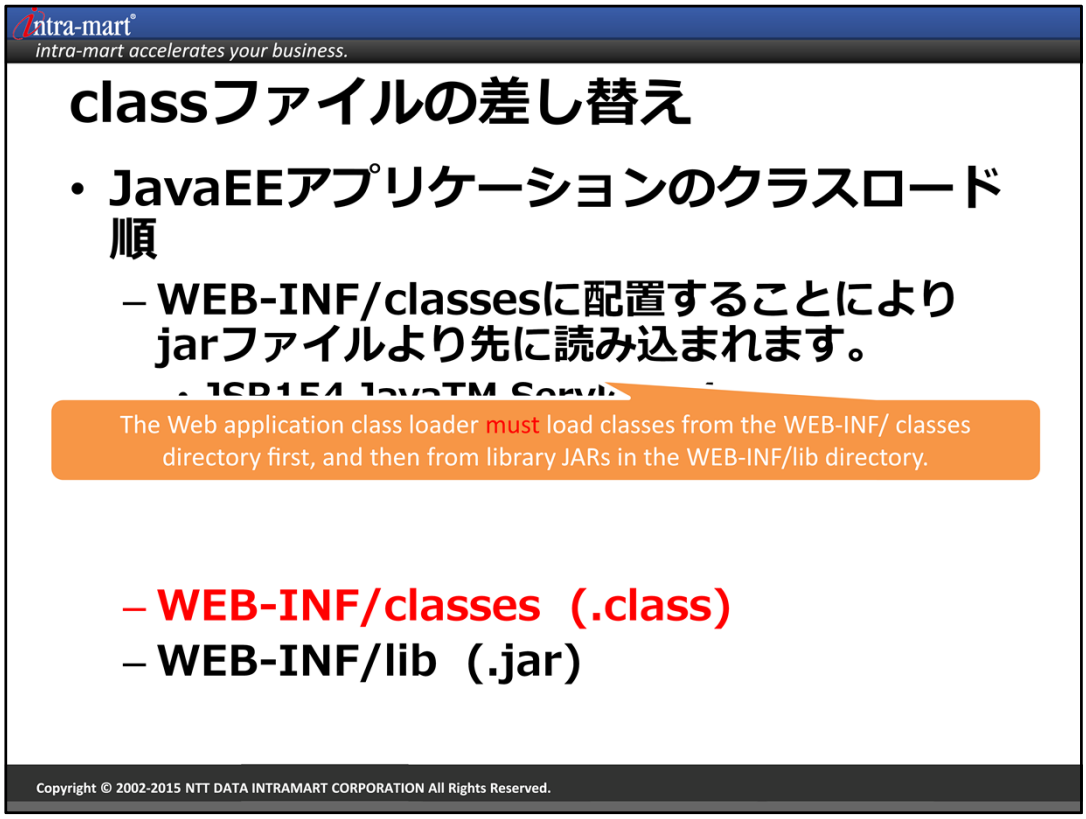
拡張ポイントに対する設定ファイルを記述することにより任意の処理を埋め込むことが可能となります。

拡張ポイント以外に、iAPではAPIの内部実装においてServiceLoaderを利用しています。
ServiceLoaderは、APIの処理に対する実装を差し込めるような仕組みとなっております。
その為、特定のAPIの動作を変更したい場合において、元のソースコードを修正すること無く実装の差し込みを行うことが可能です。



上記の方法以外で、直接ソースコードをカスタマイズする必要がある場合には、classesディレクトリにクラスファイルを配置します。

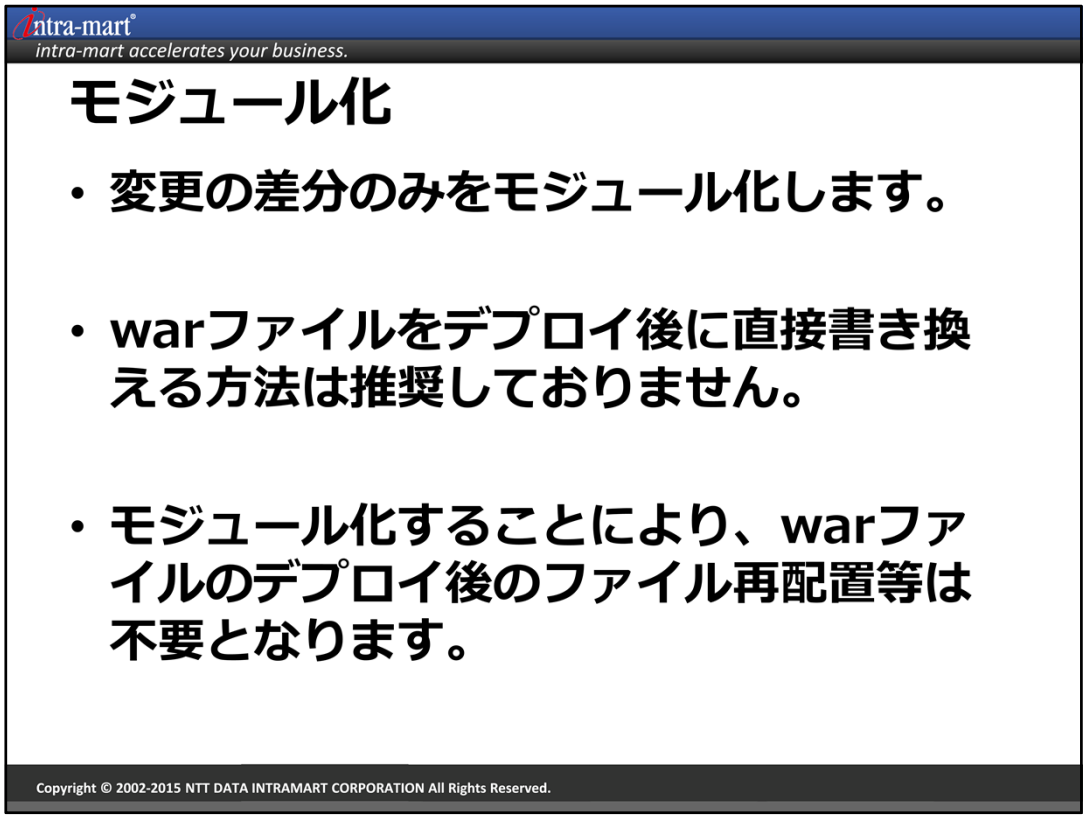
これは、Resin上でのクラスをロードするディレクトリの順序です。



Servletの規約により、classesディレクトリは、libディレクトリより先にクラスがロードされます。

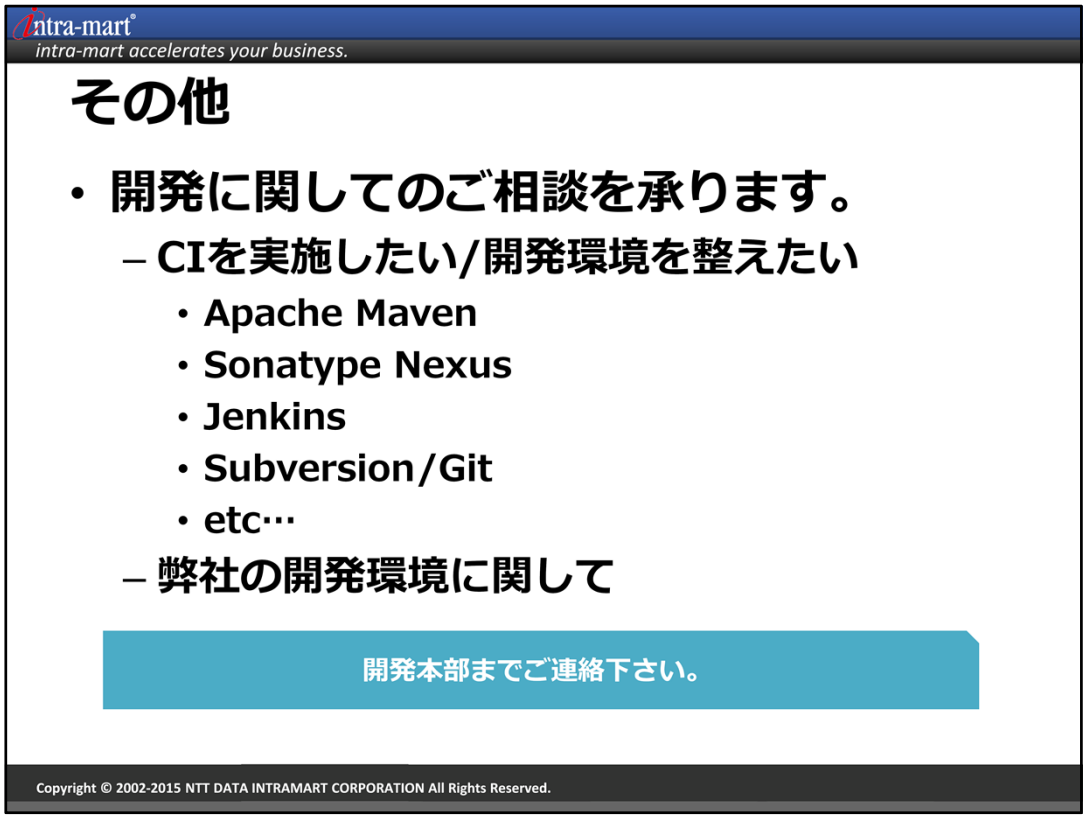
その為、カスタマイズしたクラスファイルをclassesに配置することによりその部分のみ差し替えることが可能です。

弊社では、以前Strutsの脆弱性の際に公開したパッチや、特定のお客様向けにお出ししている個別パッチなどでこの差し替えを行っております。



ここまでお話しましたカスタマイズ方法を利用し、iAPのカスタマイズを実施されましたらそのカスタマイズ部分の差分のみをユーザモジュールとしてモジュール化します。

Warファイルをデプロイ後、カスタマイズ部分のコードを直接配置される等の方法に関しては推奨しておりません、こちらの方法を実施された場合、変更の追随が困難となる、分散環境の自動スケール等が行えない等のデメリットが存在します。



iAPにおけるカスタマイズ方法の紹介でした。

また、弊社では、パートナー様の開発におけるご相談を承っております。

今回お話させて頂いた中で、一部キーワードとして出てきましたApache Maven、Mavenを利用した自動ビルドでしたり、

Jenkins等と組み合わせたCIの実現方法等を既に何社かのパートナー様とお話させて頂いております。

また、弊社内における開発の自動化等も直接ご覧頂きながらご紹介できますので興味お持ちになられましたら是非お声がけください。

以上がiAPにおける開発に関する内容となります、ご清聴有難うございました。

intra-mart®
intra-mart accelerates your business.

お問い合わせ

intra-mart
Open & Easy

株式会社NTTデータイントラマート

本社：
〒107-0052
東京都港区赤坂4-15-1 赤坂ガーデンシティ5階
TEL：０３－５５４９－２８２１
FAX：０３－５５４９－２８１６

URL：http://www.intra-mart.jp/