intra-mart WebPlatform/AppFramework
Ver.7.0

イントロダクション

2010/11/30 第5版

❖ 変更履歴

変更年月日	変更内容
2008/07/07	初版
2008/07/09	第2版 3.6 旧バージョンとの互換性についての記述誤りを訂正 3.6.3 旧リビジョンとの互換性（Ver.6.1→Ver.7.0）での記述誤りを修正
2008/08/22	第3版 2.6.4 Sersar2での完全なHotDeployの機能の提供での誤字を修正 3.1.2.4 アプリケーション共通マスタ Colummの画像を修正
2008/09/30	第4版 2.6.2 ログイン画面のヘッダ・フッタおよびトップバー画像のカスタマイズの記述誤りを修正
2010/11/30	第5版 3.1.5.6 IM-SonicESBの記述誤りを修正

Contents

第1章　intra-martとは　1

1.1　intra-martの全体構成　2

1.1.1　製品構成　3

1.2　intra-martの特長　4

1.2.1《特長1》高い開発生産性を実現する統合型フレームワーク　5

1.2.2《特長2》ビジネスの変化に柔軟に対応できるSOAシステム統合基盤　8

1.2.3《特長3》高速で信頼性のあるOSSアプリケーションサーバ機能も装備　9

第2章　新機能紹介　10

2.1　BPM　11

2.2　ポータルモジュール　12

2.3　ワークフローモジュール　14

2.3.1　フロー作成時の操作性向上　14

2.4　Webサービス　15

2.5　ログイングループ管理者権限の委譲　16

2.6　その他の新機能　17

2.6.1　ログ機能　17

2.6.2　ログイン画面のヘッダ・フッタおよびトップバー画像のカスタマイズ　17

2.6.3　マスカット2.0.0対応　18

2.6.4 Seasar2での完全なHotDeployの機能の提供　18

第3章　intra-martの標準機能　19

3.1　im-BizAPI（Java業務コンポーネント群）概要　20

3.1.1《利用方法1》JavaEE開発モデルの利用　20

3.1.2《利用方法2》スクリプト開発モデルの利用　21

3.1.3 Java業務コンポーネント群（im-BizAPI）　21

3.1.4　標準画面の作り方（共通画面デザイン）　43

3.1.5　エクステンションシリーズ　47

3.2　intra-martのアプリケーション開発概要　55

3.2.1　スクリプト開発モデルによるアプリケーション開発　56

3.2.2 JavaEE開発モデルによるアプリケーション開発　58

3.3　intra-martのシステムアーキテクチャー　63

3.3.1 intra-mart WebPlatformのシステムアーキテクチャー　63

3.3.2 intra-mart AppFrameworkのシステムアーキテクチャー　65

3.4　intra-martのその他の特徴　67

3.4.1　多言語対応　67

3.4.2　運用の容易性　68

3.4.3　強固なセキュリティ　71

3.4.4 ASP型アプリケーションの構築　71

3.5　intra-mart「eBuilder」　72

3.5.1 Script Producer　72

3.5.2 FrameworkProducer　73

3.5.3 FrameworkProducerのソースコード自動生成、ビジュアルな画面定義機能　73

3.5.4　マスカットIDEによるビジュアルな画面定義機能　75

3.6　旧バージョンとの互換性　76

3.6.1 APIの互換性　76

3.6.2　データベース構造の互換性　76

3.6.3　旧リビジョンとの互換性（Ver.6.1→Ver.7.0）　76

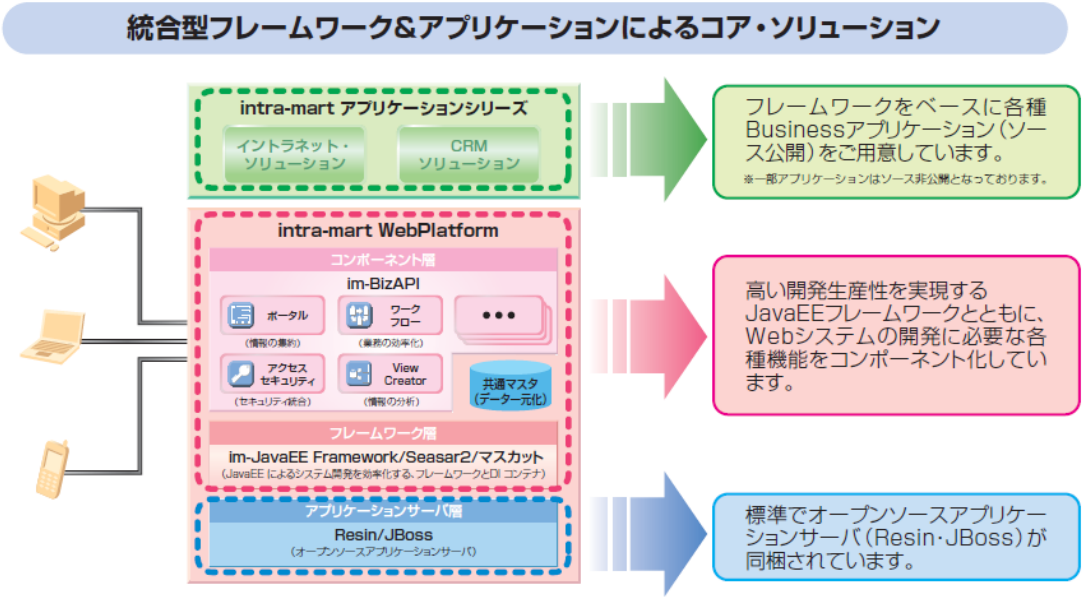
3.6.4 JumpUpモジュール（Ver.3、4→Ver.5.1~Ver.7.0）　76

intra-mart WebPlatform/AppFramework

第1章 intra-martとは

1.1 intra-martの全体構成

intra-martでは企業のWebシステムを実現するために、各種のソリューションを提供しています。



intra-mart WebPlatform (フレームワークベースの次世代型Webアプリケーションサーバ)	intra-mart WebPlatformは、フレームワークを利用することで、複雑なWebアプリケーションを簡単に短期間で開発することができます。フレームワークには、さまざまなJava業務コンポーネント(アクセスセキュリティ・モジュール/ワークフロー・モジュールなどを総称して「im-BizAPI」と呼びます)とJavaEEフレームワーク(im-JavaEE Framework)から構成されています。さらに、JavaEEベースの開発と併用して、スクリプト開発モデルも利用できるため、Webシステム開発の敷居を大きく下げることができます。また、フェールオーバー機能やラウンドロビンによるロードバランス機能などの高度な信頼性を提供するJavaEEアプリケーションサーバの機能(Resin/JBoss)も標準搭載されています。
intra-mart AppFramework	intra-mart WebPlatformのフレームワークを他社アプリケーションサーバ(IBM WebSphere、BEA WebLogic、NEC WebOTXなど)でもご利用になれるフレームワークだけの製品です。JavaEEアプリケーションサーバ上でのWebシステム開発の生産性を大きく向上させることができます。
intra-martアプリケーションシリーズ	intra-martアプリケーションは、強力なフレームワークにより構築された統合型Web業務アプリケーションシリーズです。すべてのソースコードが公開されており、ユーザは自由にカスタマイズして機能を拡張していくことができます。さらに、各アプリケーションの会社/組織情報、顧客情報など基本的な情報が連動して動作します。これらのアプリケーションでは、既存の基幹システムと連携した全社規模の情報共有を実現することで、経営課題を短期間で達成していくことができます。 各アプリケーションのシリーズ製品は以下の2つのソリューションに分類されます。
イントラネット・ソリューション	M&A、アウトソーシングニーズの高まりにより増大する「間接部門のコスト低減」と「全社規模での情報共有(コラボレーション)」を実現するWebイントラネット構築ソリューション。全社規模のイントラネットを実現します。
CRMソリューション	折衝情報、サポート情報、クレーム情報などの顧客情報の全社共有は、企業の最重要テーマとなります。CRMソリューションは、顧客情報を中心としたエンタープライズポータルを実現します。

intra-martでは、イントラネット、CRMなど各種Webアプリケーション製品により、力強くトータルにバックアップしていきます。またこれらのアプリケーションシリーズはすべてソースコードが公開されているため、ソースコードに対してユーザ自身で簡単にカスタマイズできるなどの特徴があります。
本書では、intra-mart製品の中核をなす基盤ツールであるintra-mart WebPlatform/AppFrameworkについて説明します。
その他、intra-martアプリケーションの詳細については、それぞれのマニュアル等を参照してください。

1.1.1 製品構成

intra-mart WebPlatform/AppFrameworkには、以下のような3つの製品が用意されています。

機能	スタンダード	アドバンスト	エンタープライズ for BPM
アプリケーションサーバ*1	○	○	○
im-BizAPI	○	○	○
アクセスセキュリティ機能	○	○	○
ポータル	○	○	○
スクリプト開発モデルによるアプリケーション構築	○	○	○
JavaEE開発モデルによるアプリケーション構築	○	○	○
EJBサーバ機能*1	○	○	○
ワークフローモジュール		○	○
FormatCreator		○	○
BPM|Server BPM|designer（Intalio）			○
ESB Mule			○

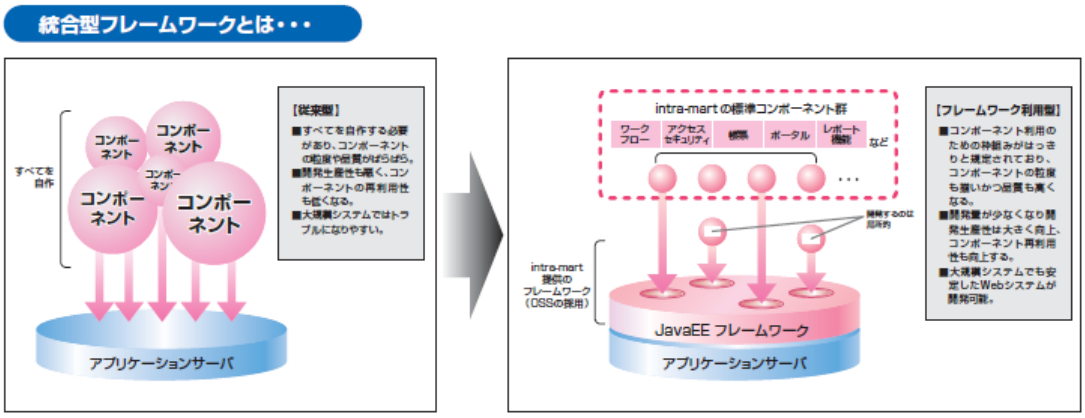
*1 intra-mart AppFrameworkには付属しない機能です。

1.2 intra-martの特長

需要の高まるWebシステムの開発には、以下のような課題があります

- ❖ **短納期・低コストへの要求（IT 投資に対する厳しい評価）**
- ❖ **オープンソースソフトウェア（OSS）の台頭（特定ベンダーからの離脱）**
- ❖ **ビジネス変化に柔軟に対応するシステムへの期待**

これらの課題に対応するため、システム開発はフルスクラッチ（ゼロからの開発）から、コンポーネントやOSSの組み合わせ利用型へと変化しています。
そこで、開発期間の短期化、開発コストの低減に加え、標準化の推進・品質の向上等にもメリットがあることから、フレームワークやコンポーネントを組み合わせた「統合型フレームワーク」に対する要求が高まってきました。



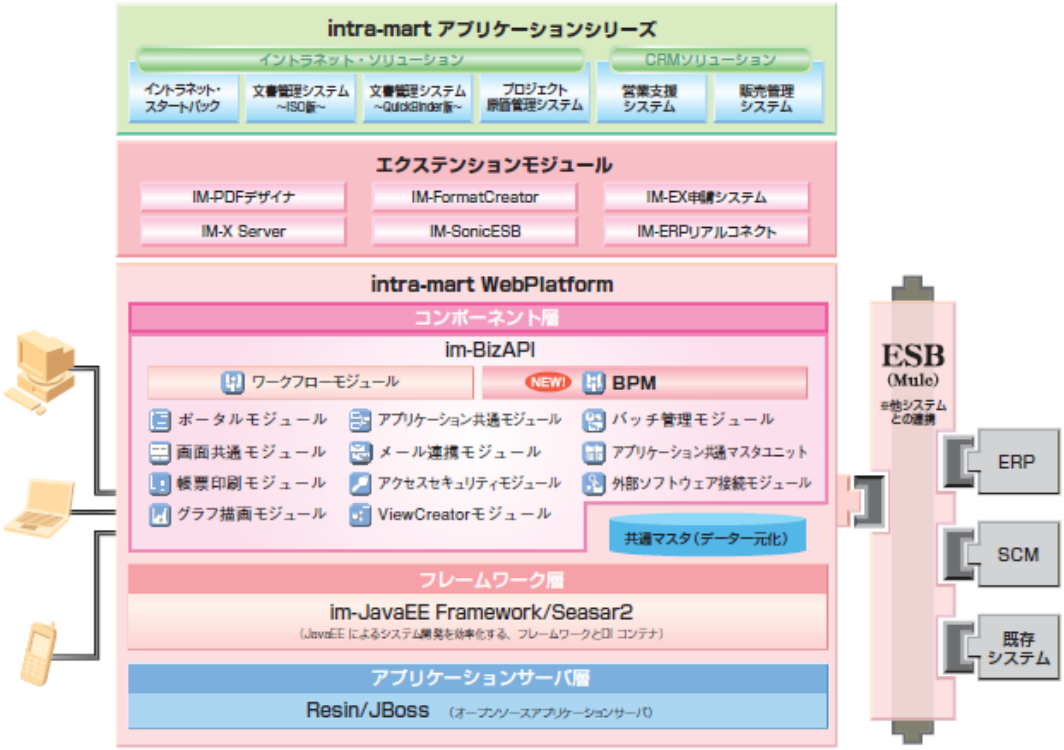
この統合型フレームワークを、全社共通のプラットフォームとして有効活用していくことが、激しい競争を勝ち抜くための企業戦略として欠かせません。
intra-martのフレームワークを利用することで、複雑なWebアプリケーション開発を容易に、短期間で実現するとともに、ビジネス変化にも柔軟に対応できる保守性の高いWebシステムを構築することができます。

1.2.1 《特長1》高い開発生産性を実現する統合型フレームワーク

統合型フレームワークは、さまざまなJava業務コンポーネント（im-BizAPI）と、JavaEEフレームワーク（im-JavaEE Framework）から構成されています。
さらにJavaEE開発モデルとスクリプト開発モデルを併用できるため、Webシステム開発の敷居を大きく下げることができます。

1.2.1.1 数多くのJavaコンポーネント群（im-BizAPI）

JavaEE対応Webアプリケーションサーバによるシステム開発では、一般的にJavaクラスやEJB（Enterprise Java Beans）コンポーネントなどのソフトウェア部品を活用しながら開発生産性を高めていくことができる特徴があります。
しかし、このソフトウェア部品の開発には特殊なスキルが必要とされ、依然として「しきい」が高いのが現状です。
従来ゼロから開発したり、他社製品を組み合わせたりしないと実現しなかった複雑な機能を、フレームワークで用意されているJava業務コンポーネント群を利用することで、大規模なWebシステムでも短期かつ高品質に構築できます。
またソースコードも公開されているので、自社独自の機能拡張フレームワークに発展させることができます。

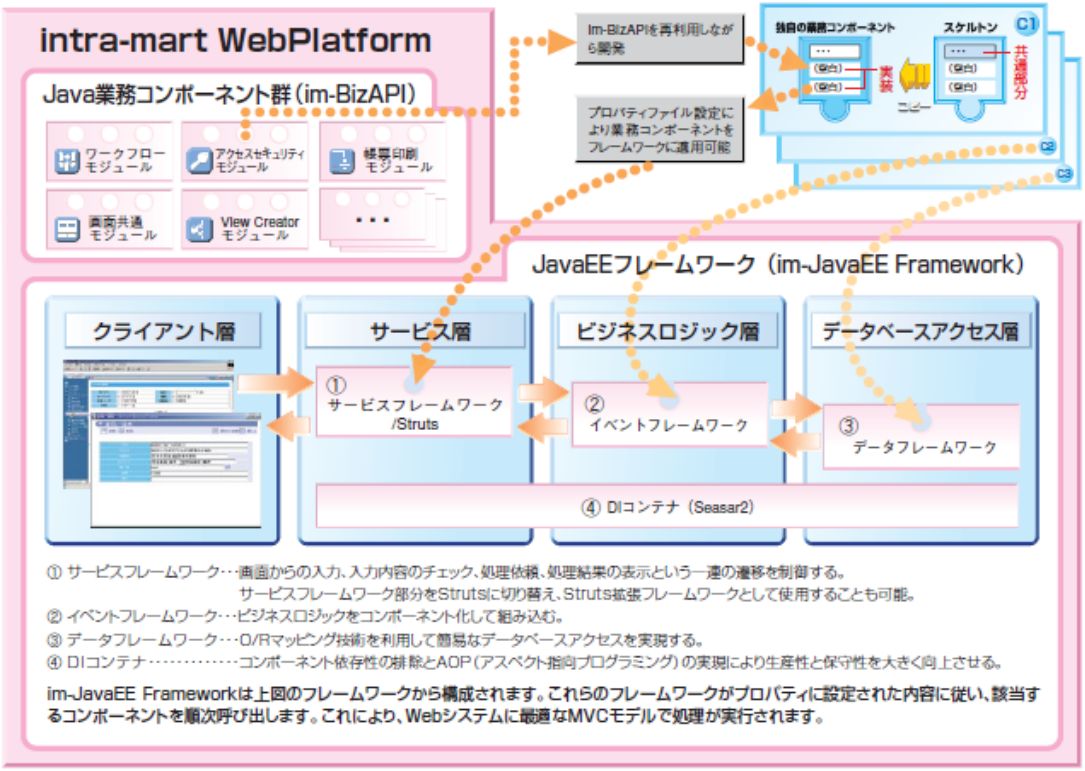


※「intra-mart WebPlatform」にはオープンソースアプリケーションサーバ（Resin/JBoss）が同梱されます。他社アプリケーションサーバ（WebLogic/WebSphere/WebOTX）上で利用する場合はWebアプリケーションサーバを含まないフレームワーク製品「intra-mart AppFramework」をご利用ください。

標準で用意されている各種Javaコンポーネントを組み合わせることで、要件に合わせた柔軟なカスタマイズが可能です。

1.2.1.2 JavaEE開発のフレームワーク(im-JavaEE Framework)

JavaEE開発モデルの生産性を向上させるため、JavaEE開発で必要になる共通的な処理はすべてフレームワーク（im-JavaEE Framework）として用意し、開発者に委ねられる箇所はコンポーネントを作成してもらう形態となります。
特に、このフレームワークを利用するコンポーネントについては、雛型となる"スケルトン"を用意しておくことで、さらに生産性を高めることが可能となります（共通部分はあらかじめ実装されているので、開発者は独自のコンポーネントを作成する場合、スケルトンをコピーし、実装されていない箇所を埋めてコンポーネントを作成することになります）。作成されたコンポーネントはプロパティ設定によりフレームワークにあてはめていくことで動作します。



スケルトンを利用したコンポーネント開発のため、JavaEE開発の前提知識がなくても完成したシステムはJavaEEアーキテクチャの推奨型となり、MVCモデルの実現が容易になります（プログラム構造が統一できるため、コンポーネントの再利用性が高まり生産性および保守性が向上します）。また、コンポーネント群とプロパティファイルの設定をベースとした構造になるため、柔軟なカスタマイズや機能追加が可能なシステムとなります。

■JavaEE開発モデル

JavaEE（Java2 EnterpriseEdition）とは米サンマイクロシステムズが1999年9月に発表したプラットフォームで、Servlet、JSP（JavaServerPage）、EJB（EnterpriseJavaBeans）などから構成され、MVCモデル（Model-View-Controller）でシステム構築をおこなっていきます。特に高トランザクションが集中する処理に適し、さらに業務コンポーネントの再利用が可能になるなどのメリットがあります。

- intra-mart WebPlatform/AppFrmeworkは、オープンソースDIコンテナであるSeasar2を組み込んでいます。
- Seasar2の機能であるHOT deployも使用することができます。

1.2.1.3 スクリプト開発モデルによる開発も同時にサポート

intra-mart WebPlatform/AppFrameworkでは、世界で初めて2つのWebアプリケーションモデルを同時にサポートしています。「JavaEE開発モデル」と「スクリプト開発モデル」という２つの開発モデルは、それぞれ特徴を持ち、用途に応じて使い分けることができます。また、2つのモデルで開発したWebアプリケーションを同一のシステム内に混在させて実行することもできます。

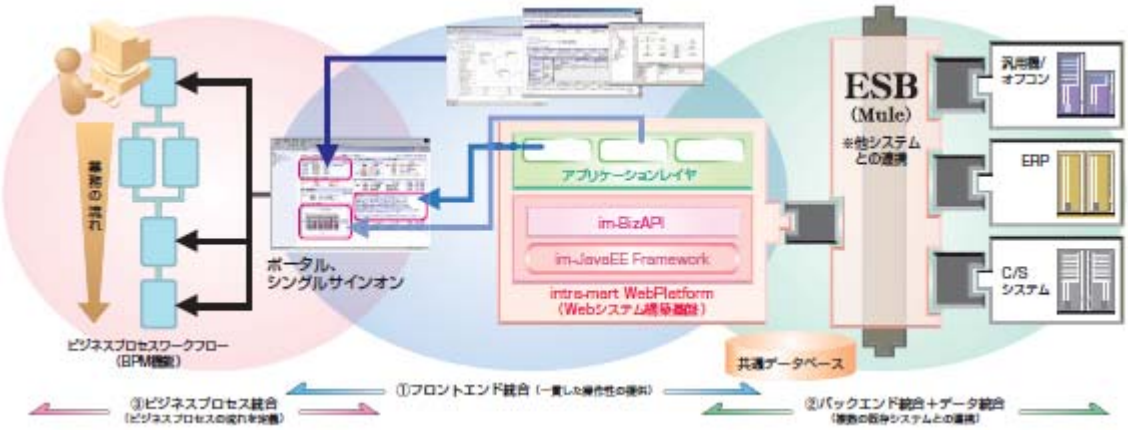
どちらの開発モデルもアクセスセキュリティ・モジュール、ワークフロー・モジュールなど、intra-martが標準で用意しているJavaコンポーネント群（im-BizAPI）を利用することができますので、高い生産性を誇ります。そのため、開発予算が限られ短期間開発の必要なWebシステムはスクリプト開発モデルを主体に開発を進め、その中で異なるプロジェクトでも作成したコンポーネントの再利用が可能な部分を切り出してJavaEEベースで開発するといった、現実的かつ柔軟な開発スタイルをとることができます。この方式により、JavaEEベースの開発モデルでの問題（クラス設計などの高度なスキルの必要性やJava技術者の確保などの面から敷居が高く開発費用が増大する）を解決することが可能となります。

■スクリプト開発モデル

ＰＨＰやＲｕｂｙに代表されるようなスクリプトを主体にした開発モデルです。intra-martでは、普段ホームページの作成に利用しているHTMLと（サーバサイド）JavaScriptを利用してデータベースと連動したWebシステムの開発が可能な作りにしています。そのため初心者でも始められる簡易性により要員のシステム教育コストを大きく削減することができます。これまで簡単なホームページしか作成したことがなかった方でも約２週間～１ヶ月ほどでマスタしてWeb業務画面を作成しています。
また、ホームページの作成更新と同様に簡易に開発ができますので、頻繁に更新されるような複雑なWebシステムにも柔軟に対応できるメリットがあります（さらに、JavaScriptからは、開発者のテクニカルスキルにあわせて、Java（Class、EJB）、C++、ストアドプロシージャなどが簡単に呼び出せます）。

1.2.2《特長2》ビジネスの変化に柔軟に対応できるSOAシステム統合基盤

Ｍ＆Ａや分社化、ビジネスモデルの変更など、ビジネスの変化に対して、システムも柔軟に対応できるような「変化に強いシステム」の重要性が高まっています。この考えがないと、ビジネス戦略にとって、情報システムが足かせとなってしまいます。intra-martではシステム統合基盤として、フレームワークに加え、アプリケーション共通マスタ、ポータルモジュール、シングルサインオン、ビジネスプロセスマネージメント（BPM）などの各種システム統合のための機能を用意しています。
これにより、ＥＲＰや汎用機などの基幹システムと密接に連携したＷｅｂフロントシステムの構築が既存システムを活かしながら短期間で構築できます。また、ビジネスプロセス、システム機能、ＤＢが独立した構造となるため、業務プロセスが変わっても業務システムの対応が短期間でできます。

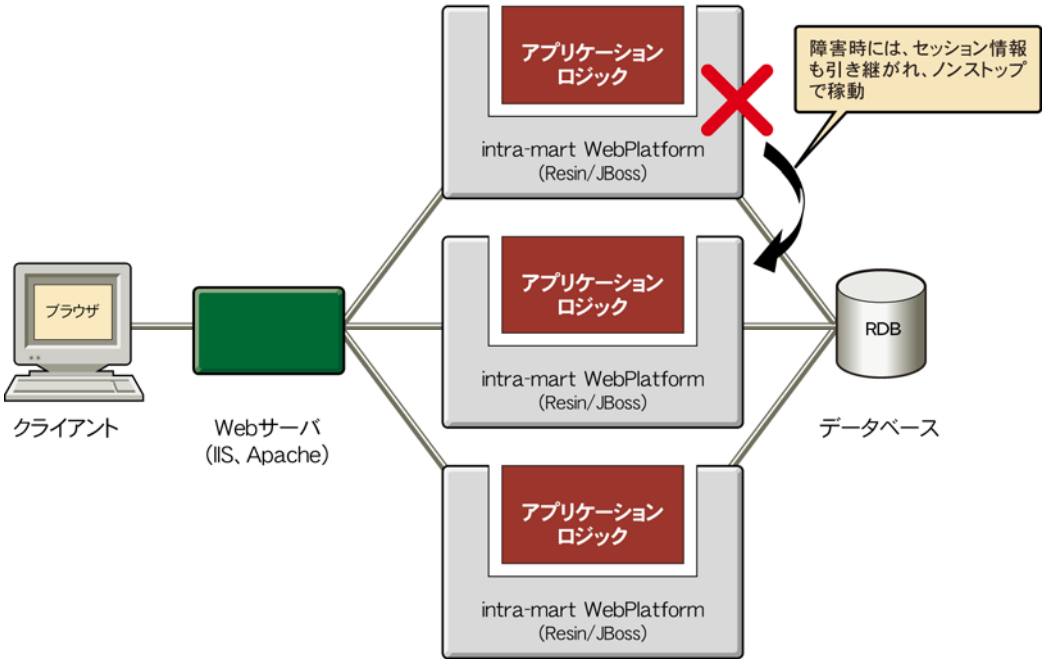


1.2.3《特長3》高速で信頼性のあるOSSアプリケーションサーバ機能も装備

intra-martのフレームワーク機能は、さまざまなアプリケーションサーバに対応していますが、intra-mart WebPlatformは、高速で信頼性の高いOSSアプリケーションサーバ機能（Resin/JBoss）も標準装備しています。そのためWebサーバ上で実行されるビジネスロジックをWebアプリケーションサーバ上で集中的に稼動し、WebサーバやDBサーバと分割した構成にすることで負荷の分散を図ることも可能です（多階層アーキテクチャへの対応）。

また、intra-mart WebPlatformでは、製品に内包されているOSSアプリケーションサーバを使用すると「ラウンドロビン機能（*1）」を利用することができます。ラウンドロビン機能は、複数のアプリケーションサーバ間でも処理の負荷分散を図ることができるため（単一のIPアドレスで）、クライアント数の増加にも柔軟に対応できるとともに、セッションフェイルオーバー（*2）などの障害対策にも優れたシステムを構築することができます。

- *1：ラウンドロビン機能とは、複数台のWebサーバをクラスタ化し、負荷を複数サーバに分散させる機能です。
- *2：セッションフェイルオーバー機能とは、複数台のWebサーバをクラスタ化し、その中のサーバが 何らかの障害（電源故障、ネットワーク障害等）でダウンするなど、サーバ機能を果たさない場合、そのサーバへのアクセスを他のサーバに振り替える機能です（セッション情報も他のサーバへ引き継ぐことができますので、完全にノンストップのシステムとなります）。



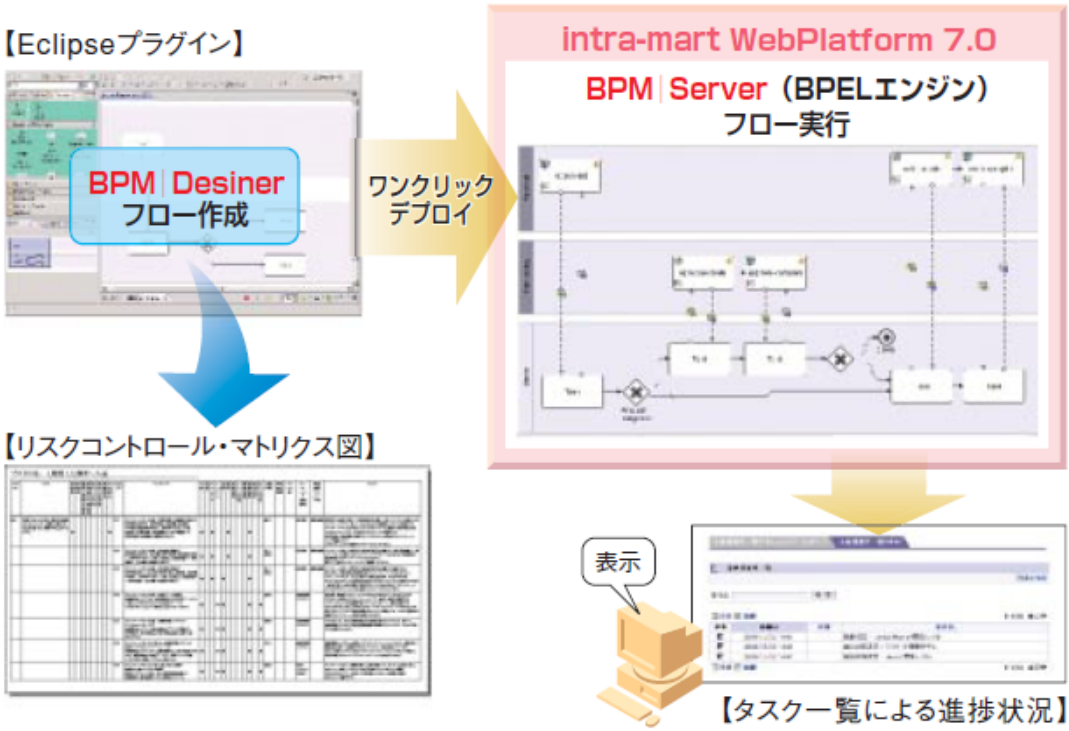
また、intra-mart WebPlatform/AppFrameworkは、HTMLやJava、JavaScriptなど、インターネット業界標準のテクノロジーのみで構築されているため、ひとつのメーカに拘束されない柔軟性の高いイントラネット構築を実現します。これらはシステムの将来性があるだけでなく、他社システムまでも包含したエクストラネットシステムを構築する際や、消費者と連動するECシステムを構築する際に、特に重要なシステム要素となってきます。

intra-mart WebPlatform/AppFramework

第2章　新機能紹介

2.1 BPM

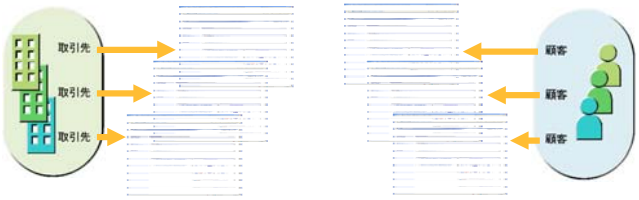
業務プロセスの可視化、連携を実現します。業務フローの定義・実行ツールとリスク・コントロール・マトリクス（RCM）図の作成機能を提供します。
BPMNに準拠しており、ビジネスプロセスをタスクという単位で、ツール上に簡単に配置していきます。
フロー記述後、ワンクリックですぐにサーバ上に展開され、実行することが可能です。



2.2 ポータルモジュール

intra-martには、ログイン初期画面（intra-martホーム画面）にあらかじめ用意したページ（ポータルアプリケーションでは、ポートレットと呼びます）をいくつかサブウィンドウとして表示し、ユーザに情報を見るように喚起させることができる機能が標準で用意されています（ポートレットを表示する画面をポータル画面と呼びます）。このポータル機能は、intra-martのポータルモジュールで実現されています。

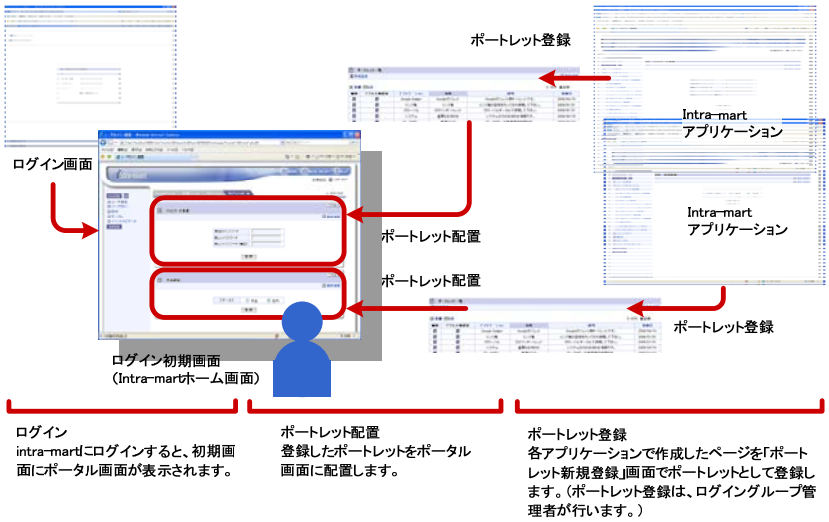
組織部署単位にポータルサイトを設定したり、個人単位の設定をすることで、組織内での情報伝達のスピードアップが図れます。



会社(顧客、取引先)単位にポータルサイトを設定することにより、
ECサイトやマーケットプレイスにおいて、サービスレベルを大きく向上させることができます。

〈会社・組織、ユーザごとのポータル〉

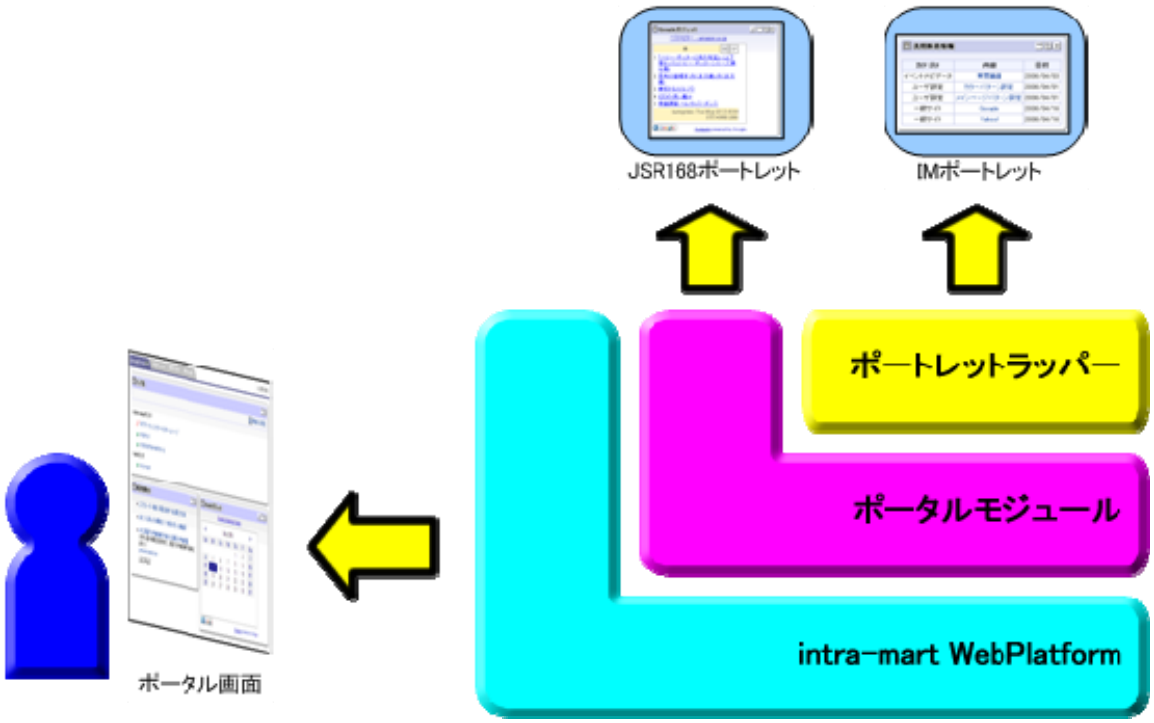
intra-mart上のアプリケーションごとに表示させたい情報のページを作成し、ポートレット管理機能を利用して登録することによって、ポートレットとして利用することができるようになります。
このようにして作成したポートレットは、ポータル画面上に自由にレイアウトして利用することができます。また、intra-mart上のアプリケーションだけでなく、Javaの標準であるJavaポートレットもポートレットとして登録し利用することが可能です。



● 詳細については、「ポータルの各種操作ガイド」を参照してください。

■JSR168対応

ポータルモジュールはJSR168（ポートレットに関する標準仕様）に対応しています。
intra-martのポータルでJSR168に対応したポートレットを利用するためには、システム管理者がポートレットのＷＡＲファイルを事前にintra-martに登録し、アプリケーションサーバにデプロイします。
その後、ログイングループ管理者がintra-martに登録されているポートレットアプリケーションの中からポータル画面で利用したいポートレットを選んで登録することによって、ポータル画面でポートレットを利用することができるようになります。
ポートレット管理では、システム管理者がポートレットアプリケーションをntra-martに登録するなどポートレット管理に関する一覧表示、登録、更新、削除を行う為の機能を提供します。



Column

JSR168（ポートレットに関する標準仕様）

いろいろな企業からポータルサーバーがリリースされていますが、各社でその仕様は統一されていませんでした。その状況を改善するために定義されたのが 標準化団体 Java Community Process（JCP）において策定された、Java標準規格「Java Specification Request #168」です。
JSR168では、ポートレットはWebアプリケーションの一種として作成され、WARファイルの形で提供されます。
この標準に準拠していれば、ポータルサーバーに依存することなく、提供されたポートレットを配備・実行することができます。
intra-martは従来のポートレットと区別する為に、JSR168に対応したポートレットをポートレットアプリケーションと呼びます。

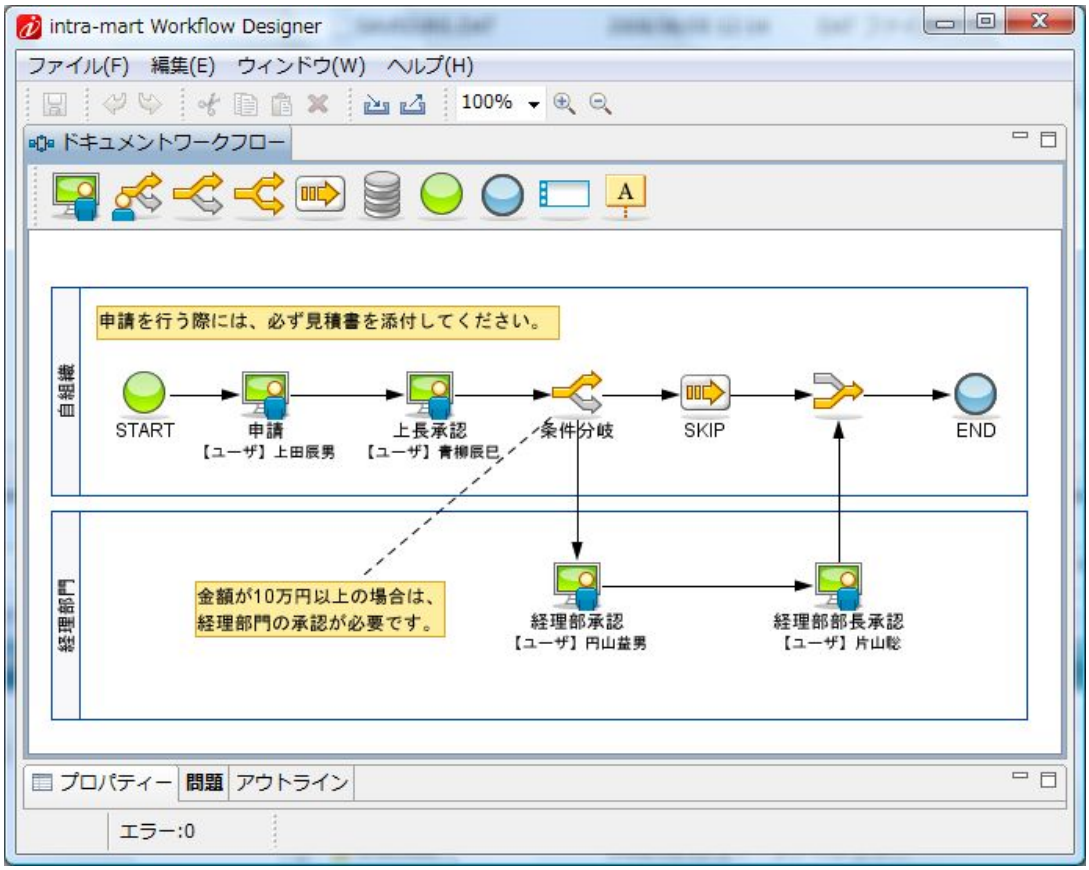
- JSR168の詳細については、「JCP」のページをご覧ください。
  http://jcp.org/en/jsr/detail?id=168

2.3 ワークフローモジュール

2.3.1 フロー作成時の操作性向上

本バージョンでは、デザイナーをEclipseRCPを利用したヘルパーアプリケーションに変更ました。
このことにより、操作性およびユーザインタフェースが大幅に向上しました。
このデザイナーでは従来の機能に加えて以下の機能が拡張されています。

- 新デザインによるユーザインタフェースの向上
- フロー全体を表現するアウトラインビュー
- フロー作成におけるアンドゥ・リドゥ機能
- フローのコピー・ペースト機能
- フローへの注釈(Note)の挿入機能
- フロー内の各業務をカテゴライズするスイムレーンの描画機能
- 逐次フローの整合性を検証するリアルタイムフローチェック機能
- フロー情報を一時的にクライアントに保存する機能
- フローを画像ファイルとして出力するエクスポート機能
- 各種検索画面(ユーザ、ロール、組織、パブリックグループなど)の操作性の向上



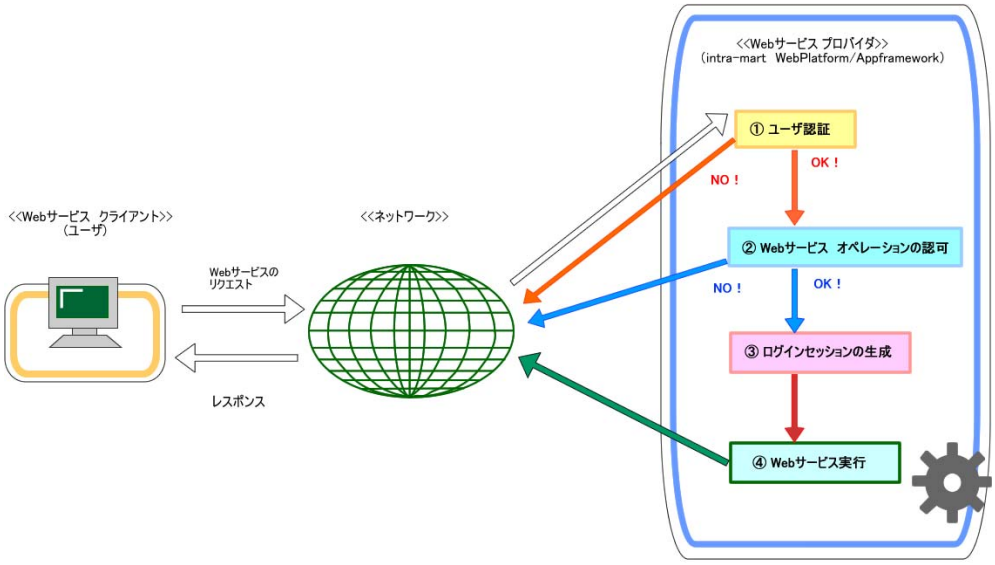
- ワークフロー機能は「アドバンスト」および「エンタープライズ for BPM」のみの機能となります。
- 詳細については、「ワークフロー 管理者操作ガイド」を参照してください。

2.4 Webサービス

intra-martでは、Apache Axis2を用いることによりSOAPベースのWebサービスを提供することが可能です。
Webサービスを提供する際に、認証と認可の2重のセキュリティを確保しますので、より安全なWebサービスを提供することが可能になります。
また、サーバサイドJavaScriptでWebサービス・クライアントを簡単に作成することも可能です。

■Webサービスアーキテクチャ

Webサービス・クライアント(ユーザ)がWebサービス・プロバイダ(intra-mart WebPlatform/Appframework)に対し、Webサービスを要求します。
このとき、Webサービス・プロバイダ側では① **ユーザ認証**を行います。Webサービス・クライアントが認証されると、処理が② **Webサービス オペレーションの認可**へ移ります。
認可されると、③ **ログインセッションの生成**、そして、④ **Webサービスの実行**へ処理が移ります。
その後、Webサービスの実行結果がWebサービス・クライアントに返却されます。



＜Webサービス実行の流れ＞

Column

Webサービス用語

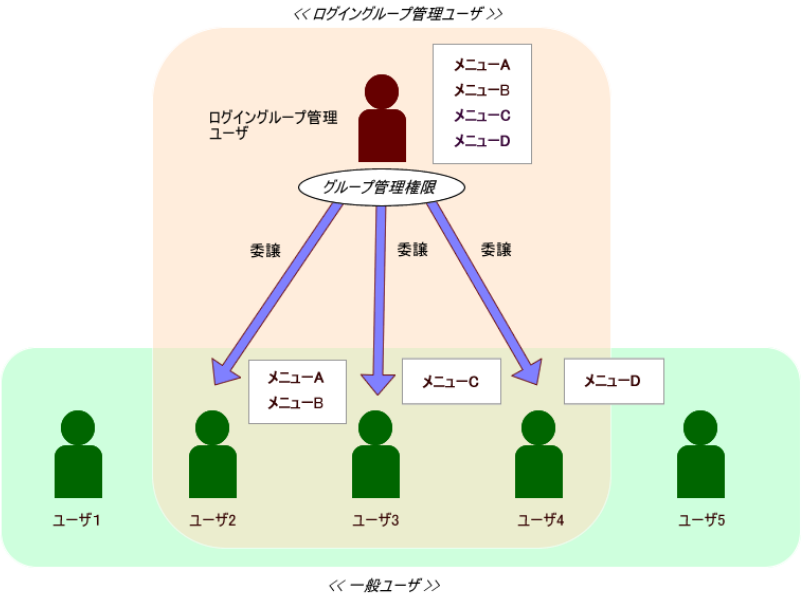
認証
Webサービス・クライアントが正当なユーザであることを確認することを意味します。

認可
認証されたWebサービス・クライアントにWebサービス オペレーションのアクセスを許可することを意味します。認可はロール単位で設定できます。

2.5 ログイングループ管理者権限の委譲

本バージョンからログイングループ管理者を複数人で運用することが可能になりました。
ログイングループを複数のユーザで管理することによって、内部統制による管理がより強化されます。

ログイングループ管理者が一般ユーザにログイングループ管理権限を委譲することにより、一般ユーザによるログイングループの管理が可能となります。
この機能によりログイングループ管理者は一般ユーザに管理メニューの使用権限を割り当てる（委譲する）ことが可能になります。



<ログイングループ管理権限ユーザ設定画面>

このように、ログイングループの管理メニューを複数のユーザで管理することで、内部統制にそった運用が可能となります。

- 詳細については、「グループ管理者　操作ガイド」を参照してください。

2.6 その他の新機能

2.6.1 ログ機能

一般的なログライブラリ（logback）の採用により、ログ機構がシンプルになりました。
また、ログの種類ごとに設定ファイルを作成し、管理することができるようになります。
ログ機能で提供可能になった機能は、以下の通りです。

- ログ API の追加により、開発者が容易にログを出力できます。
  ログ API は、JavaEE 開発モデルおよびスクリプト開発モデルで利用可能です。
- 新しい種類のログファイルを容易に作成できるようになります。
- ログに以下の 5 段階のレベル（ERROR、WARN、INFO、DEBUG および TRACE）を指定することが可能です。
- エラーログに限らず、全てのログで Exception を別ファイルへ出力することができます。
- ログローテイトの際にファイルを圧縮（zip、gz）することができます。
- ログの出力先（コンソール、ファイルおよびメール）が選択できます。

● 設定に関しては、「ログ 設定ガイド」を参照してください。

2.6.2 ログイン画面のヘッダ・フッタおよびトップバー画像のカスタマイズ

ログイン画面およびメイン画面に対して、設定による画面のカスタマイズが可能です。
これにより、ユーザ固有の画面を作成することができます。
設定によりカスタマイズ可能な部分は以下の通りです。

- ログイン画面のヘッダー部分およびフッター部分
- メイン画面のトップバー内のロゴアイコン

〈ログイン画面〉

〈メイン画面のトップバー〉

● 設定に関しては、「アクセスセキュリティ仕様書」を参照してください。

2.6.3 マスカット2.0.0対応

マスカットは、オープンソースであるAjaxを用いたリッチクライアントフレームワークです。
マスカットでは、Ajaxのプログラミングをすることなく、リッチクライアントを構築することができます。
また、前バージョンに比べ開発環境が大幅に強化されています。
intra-martでは、本バージョンからマスカット 2.0.0にバージョンアップしています。

2.6.4 Seasar2での完全なHotDeployの機能の提供

Seasar2のHotDeployを利用することで、サーバの再起動を行うことなく、開発したコンポーネントが即座にデプロイされ利用できるようになります。
この機能により開発におけるオーバヘッドを軽減することが可能になります。
これまでは、s2Strutsにおけるアクションのみが、HotDeployの対象でした。
本バージョンからは、im-JavaEE frameworkで利用される、以下のコンポーネントについてもHotDeployの対象となります。

サービスフレームワーク	Validator ControllerObject ServiceController ServiceResult
イベントフレームワーク	Event EventListener EventResult
データフレームワーク	DAO

intra-mart WebPlatform/AppFramework

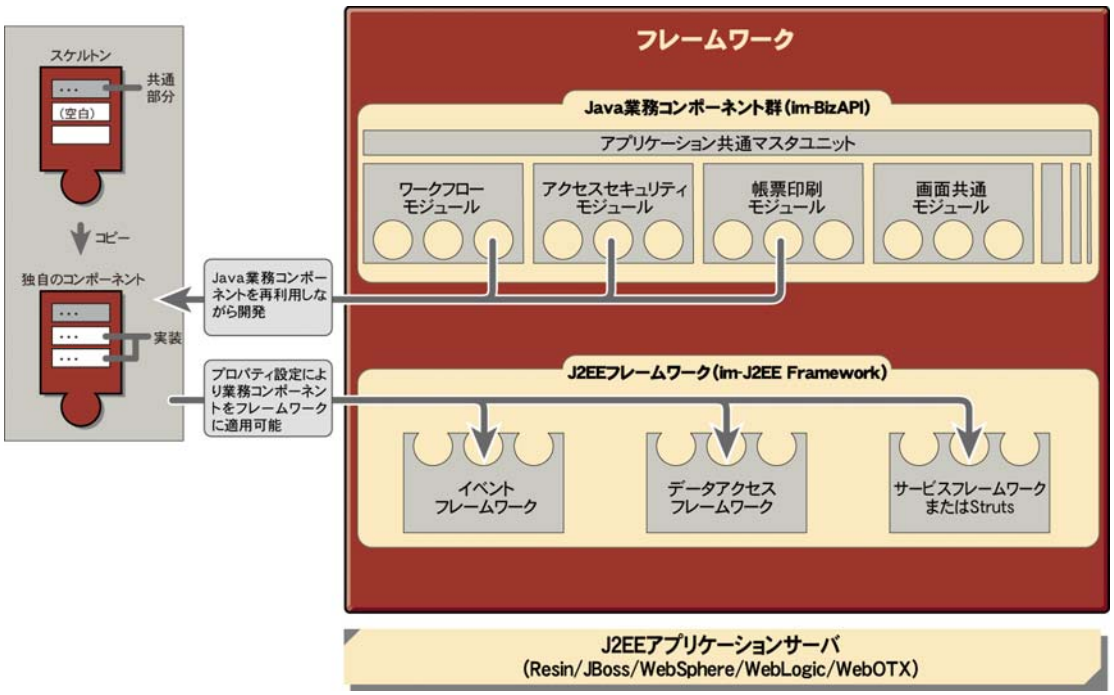
第3章 intra-martの標準機能

3.1 im-BizAPI(Java業務コンポーネント群)概要

intra-mart WebPlatform/AppFrameworkに内包されているフレームワークでは、Webシステム構築でよく利用される機能を「Javaコンポーネント」として数多く提供しており、JavaClassまたはEJBコンポーネントとして再利用が可能です（アクセスセキュリティ・モジュールやワークフロー・モジュール、ポータルモジュールなど。これらを総称して「im-BizAPI」と呼んでいます）。これにより従来ゼロから開発したり、他社製品を組み合わせないと実現できない複雑な機能を、用意されているJava業務コンポーネント群を組み合わせることで、大規模なWebシステムを短期間かつ高品質に構築できます。このim-BizAPIは、２つのアプリケーション開発モデル（スクリプト開発モデル、JavaEE開発モデル）から同様に呼び出して活用することができます。

3.1.1 《利用方法1》JavaEE開発モデルの利用

intra-mart WebPlatform/AppFrameworkに内包されているフレームワーク（下図のフレームワーク部分）には、「Java業務コンポーネント群（im-BizAPI）」だけでなく、「JavaEEフレームワーク（im-JavaEE Framework）」もあわせて用意されています。JavaEE開発で必要になる共通的な処理はすべて「JavaEEフレームワーク（im-JavaEE Framework）」内に隠蔽化されて実装されているため、JavaEEによるWebシステムの開発生産性を大きく向上させることができます。

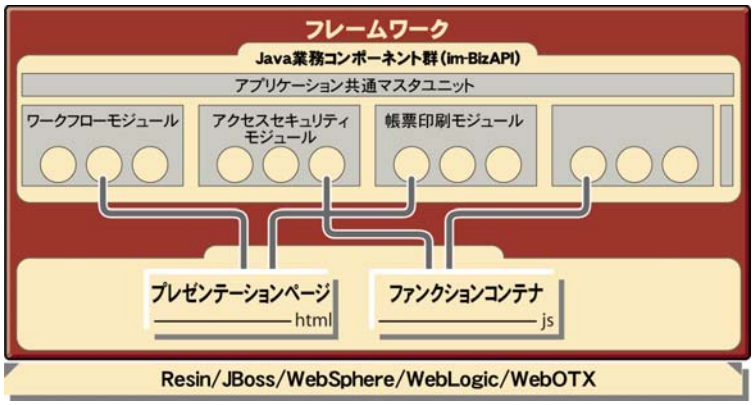


特に、このJavaEEフレームワーク（im-JavaEE Framework）を利用するコンポーネント開発については、雛型となる「スケルトン」を利用することで、さらに生産性を高めることが可能となります（共通部分はあらかじめ実装されているので、開発者は独自のコンポーネントを作成する場合、スケルトンをコピーし、実装されていない箇所を埋めてコンポーネントを作成することになります）。作成されたコンポーネントはプロパティ設定によりJavaEEフレームワーク（im-JavaEE Framework）にあてはめていくことで動作します。

- im-JavaEE Frameworkの中のサービスフレームワークの代わりにオープンソースフレームワークとして、Jakartaプロジェクトから提供されている「Struts」を利用することもできます。

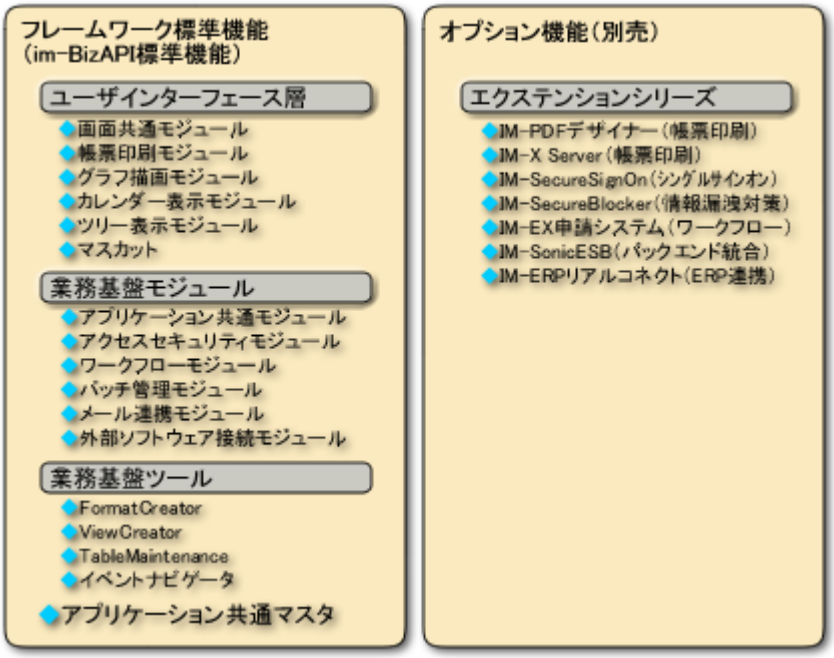
3.1.2 《利用方法2》スクリプト開発モデルの利用

スクリプト開発モデルは、取り回しに優れ、コードの作成、修正が容易と見なされる軽量プログラミング言語を採用したフレームワークを基盤とした開発モデルで、intra-martではJavaScriptを採用しています。「プログラマ負担の軽い開発」を実現し、初心者にもお勧めできる簡易性を持った開発モデルで、おもに、短期間でのWebシステム開発に向いています。JavaScriptを開発言語として採用すると、クライアントサイドおよびサーバサイド双方の開発言語を統一できるという大きなメリットが得られます。また、JavaScriptは、Ajaxが普及したことにより再び脚光をあびてきている言語でもあります。
intra-martでは、あらかじめ再利用可能な「モジュール」として用意されているJava業務コンポーネント（im-BizAPI）をプレゼンテーションページ（ユーザインタフェースを担うHTMLファイル）やファンクションコンテナ（業務ロジックを記述するJavaScriptファイル）から呼び出して利用することができます。



3.1.3 Java業務コンポーネント群（im-BizAPI）

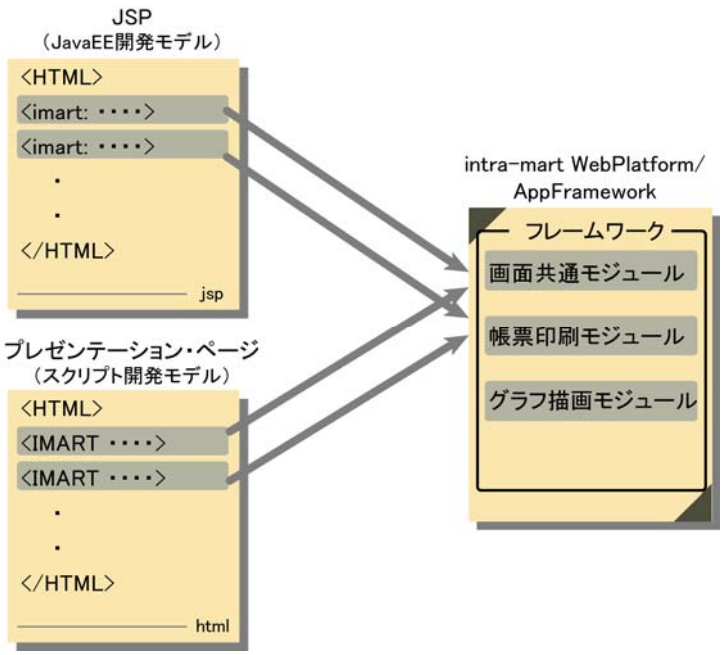
Java業務コンポーネント群（im-BizAPI）に用意されている“モジュール”は、それぞれ利用する形態が異なりますので、以下のように分類されています。各モジュールの詳細内容はAPIリストを参照してください。詳細については、「スクリプト開発モデル プログラミングガイド」および「JavaEE開発モデル プログラミングガイド」を参照してください。また、ほとんどのモジュール群のソースコードは公開されているため、モジュール自体の機能をカスタマイズして拡張することが可能です。



3.1.3.1 ユーザインタフェース層

■画面共通モジュール

WebベースでのGUI開発でよく利用される画面部品のモジュール（スプレッドシート、入力チェック、ポップアップ、タブなどのタグライブラリ）です。
それぞれのモジュールに適当なプロパティを設定して呼び出すだけで、データベースと連動したユーザーインターフェイスを簡単に作成できます。
スクリプト開発モデルではプレゼンテーションページで利用される以下のモジュール群は、すべてHTML文書の中で「<IMART>タグ」で呼び出すことができます。JavaEE開発モデルではJSPファイルの中で「intra-martタグライブラリ」を利用します。



- intra-martタグライブラリの詳細については、「API リスト」を参照してください。

■帳票印刷モジュール

これまで、帳票の印刷はイントラネットのシステム開発で一番困難と言われていました。
intra-martには、エクステンションシリーズ（別売）が用意されています。

IM-PDFデザイナー	PDFを採用することにより、より複雑な帳票形式への対応が可能となる、オプションのエクステンションモジュールです。
IM-X Server（総合帳票機能）	「IM-X Server」を利用することにより、Web帳票の電子化（PDF/XML）やサーバ大量印刷、FAX/Mail送信までトータルした帳票の課題を解決することができます。 ●きめ細かい帳票作成が簡単に行え、開発工数を大幅に削減できます。 ●オプション追加により、帳票作成の履歴管理・負荷分散/冗長化機能やFAX送信機能を付加できます。

■グラフ描画モジュール

グラフ描画モジュールを利用することで数値データをグラフ化してHTMLに表示することができます。グラフ描画モジュールには、次の5種類のグラフが用意されています。スクリプト開発モデル、JavaEEベースの両方の開発モデルに対応しています。

<利用できるグラフの種類>

- 折れ線グラフ
- 棒グラフ
- 円グラフ
- レーダーチャート
- ポートフォリオ

<グラフの表示例>

- 詳細については、「スクリプト開発モデル プログラミングガイド」および「JavaEE開発モデル プログラミングガイド」を参照してください。

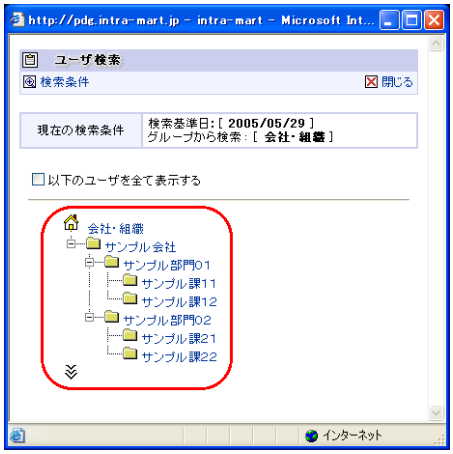
■カレンダ表示モジュール

日付の入力に使用するユニットで、曜日、休日、営業日等を管理する機能を持っています。Ajax技術を利用して実装されているので、カレンダのデータのみをサーバに問い合わせて、別レイヤとして表示されます。このため、ウィンドウ間のデータ連携等を意識することなく、使い勝手の良い画面を作成できます。会社独自の休日などの管理は、「ログイングループ管理」の「カレンダー設定」で行います。なお、カレンダーの設定は複数持つことができ、本社と工場、海外の支社など状況に応じて使い分けることができます。

- 詳細については、「スクリプト開発モデル プログラミングガイド」および「JavaEE開発モデル プログラミングガイド」を参照してください。

■ツリー表示モジュール

ツリー表示モジュールを組み込むと階層化されたデータをツリー表示することができ、階層構造の把握やメニューの選択が用意になります。



<ツリー表示ユニットの例>

- 詳細については、「スクリプト開発モデル プログラミングガイド」および「JavaEE開発モデル プログラミングガイド」を参照してください。

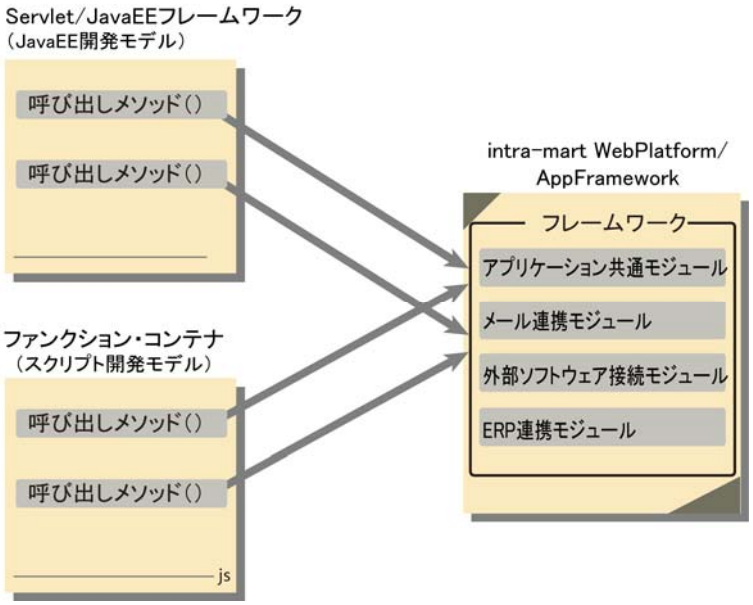
■Ajaxアプリケーション開発ツール「マスカット」

マスカットとは、Webブラウザ上で動作するAJAXベースのリッチクライアントを開発するオープンソース・フレームワークです。
intra-mart WebPlatform/AppFrameworkがサポートする２つのアプリケーション開発モデル（スクリプト開発モデル、JavaEE開発モデル）と連携することにより、リッチクライアントを容易に開発することができます。

- 詳細については「マスカット 連携 プログラミングガイド」を参照してください。

3.1.3.2 業務基盤モジュール

業務基盤モジュールには、「アプリケーション共通モジュール」、「メール連携モジュール」、「外部ソフトウェア接続モジュール」、の4つのモジュールが用意されています。これらのモジュールは、スクリプト開発モデルではファンクションコンテナからオブジェクトの形式で呼び出すことができ、JavaEE開発モデルでは、ServletからJavaClassやEJBコンポーネントとして利用することができます。



■アプリケーション共通モジュール

アプリケーション開発に必要な処理ロジックのモジュールが、使いやすくオブジェクト化されています。これらオブジェクトをビジネスロジックに組み込んで編集していくことで、ロジックの作り込みをせずに多階層アーキテクチャに基づいたWebシステムの開発を短期間で行うことが可能になります。

≪提供されるアプリケーション共通モジュールのオブジェクト概要≫

アプリケーション開発に必要な処理ロジックのモジュール（セッション管理やDBアクセスなど）が使いやすくオブジェクト化されて提供されています。このオブジェクトの利用により、ページをまたいだセッション管理を実現することができます。またフッター等に利用する会社名称や各種データベースへのログインユーザ名等の情報も、このオブジェクトから利用できます。これらのオブジェクトをアプリケーションロジックに組み込んで編集していくことで、余計なロジックの作りこみをせずに複雑なWebシステムの開発を短期間で行うことが可能になります。

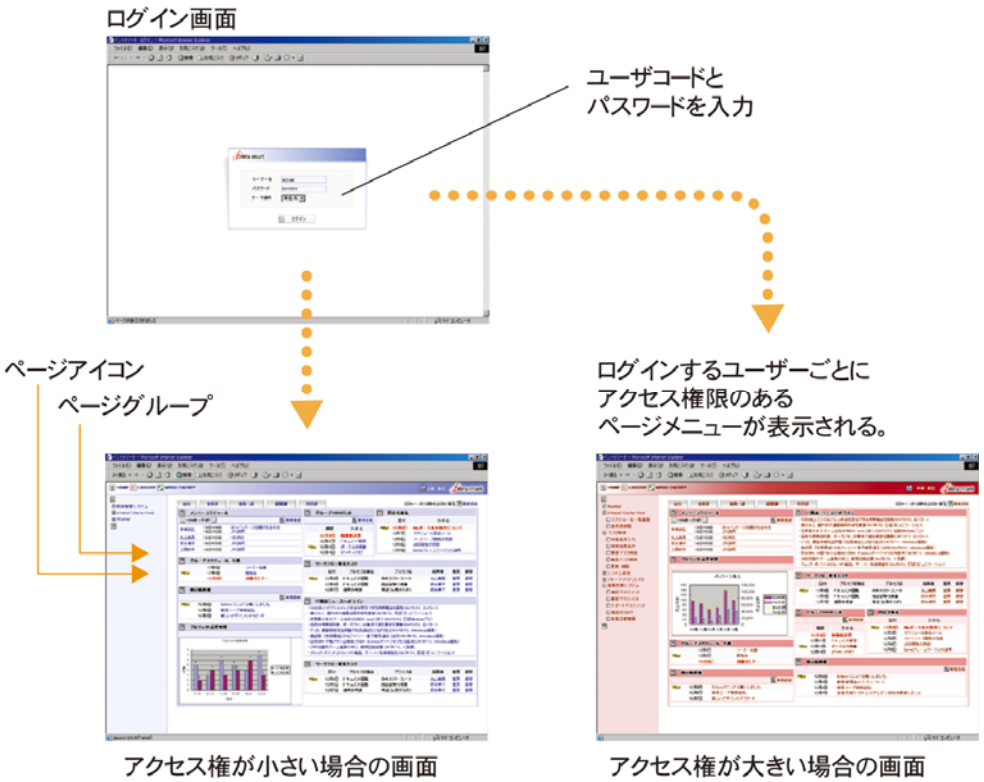
その他、アプリケーションの環境変数を含めた各設定値へのアクセス用オブジェクト、区分コードへのアクセス用オブジェクト、データベース関連の汎用オブジェクト、日付関連オブジェクト、デバック関連オブジェクト、URL管理オブジェクトなど多数用意されています。これらにより、現在接続している社員コードや、直前に表示したHTMLページ名などさまざまな情報にアクセスできます。

また、複数DBへの同時アクセスメソッドや大量の検索データに対して、指定した件数ごとに結果を画面表示する機能の検索ストリーミング、アプリケーションロック機能、XML対応モジュールなど、高度な機能が含まれています。

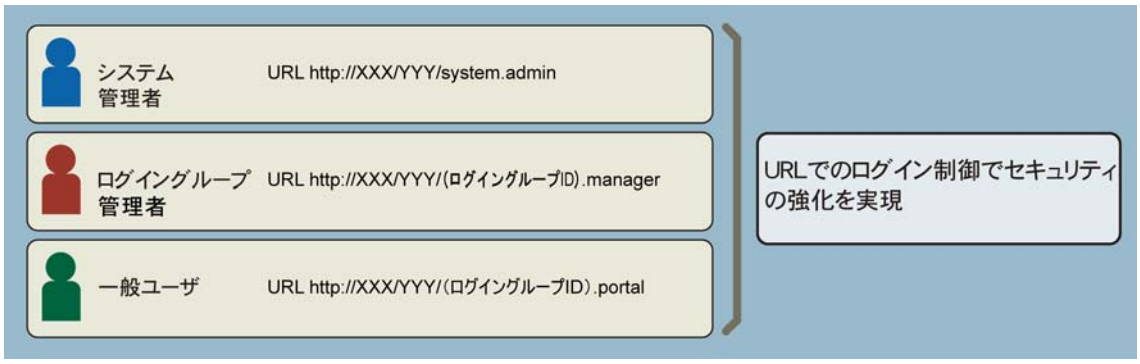
● アプリケーション共通モジュールの詳細については、「APIリスト」を参照してください。

■アクセスセキュリティモジュール

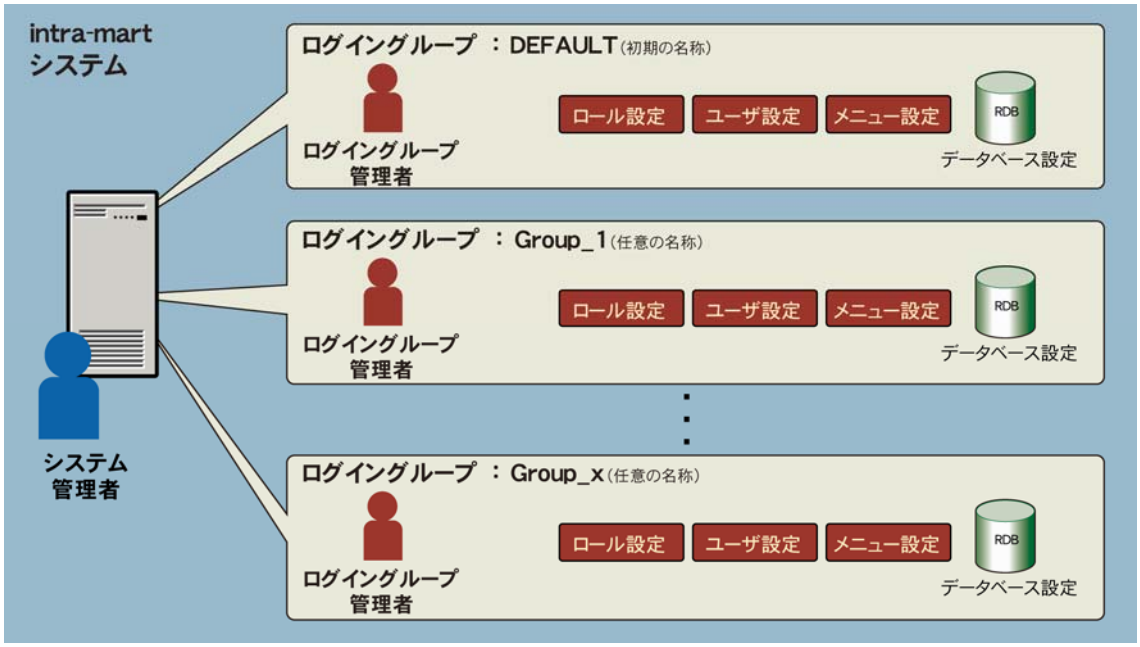
ユーザやロールなどのログイン・セキュリティ情報を操作するためのモジュールです。ログインするユーザの認証機能はもちろん、ユーザのアクセス権に応じた固有のWebページを表示することができます。たとえば、一般社員がアクセスしたときには、アクセス権限のないページはメニューにもあらわれないため、ユーザはその存在を一切意識することはありません。しかし、管理者がアクセスした場合には、同じページでも管理者が利用できるメニューまで表示されることになります。アクセスセキュリティ・モジュールを利用することで、このようなユーザに応じて内容の異なるページ構成を組むことが可能になります。
設定方法については「ログイングループ管理者 操作ガイド」を参照してください。



<ログインするユーザごとにアクセス権限のあるページメニューが表示される>



ユーザごとの言語の切り替えや画面のテーマカラーの選択が可能となりました。また、これまでユーザ情報の設定はシステム管理者がすべてを行う使用でしたが、システム管理者の配下にログイングループ管理者を設定し、ログイングループ管理者がユーザ情報の設定を行えるようになりました。



さらに、アクセスセキュリティの各機能はAPI化されており、それらを利用して、独自のメニュー画面を作成することもできます。

intra-martでは完全なSecure Sockets Layer（SSL）サポートを提供します。これにより送信されるコンテンツに対して、暗号化セキュリティをかけることが可能になります。さらに、ユーザ認証はLDAPとの連携も可能です。

標準で用意されているアクセスセキュリティモジュールのほかに、シングルサインオンを実現するエクステンションシリーズ「IM-SecureSignOn（別売）」も用意されています。

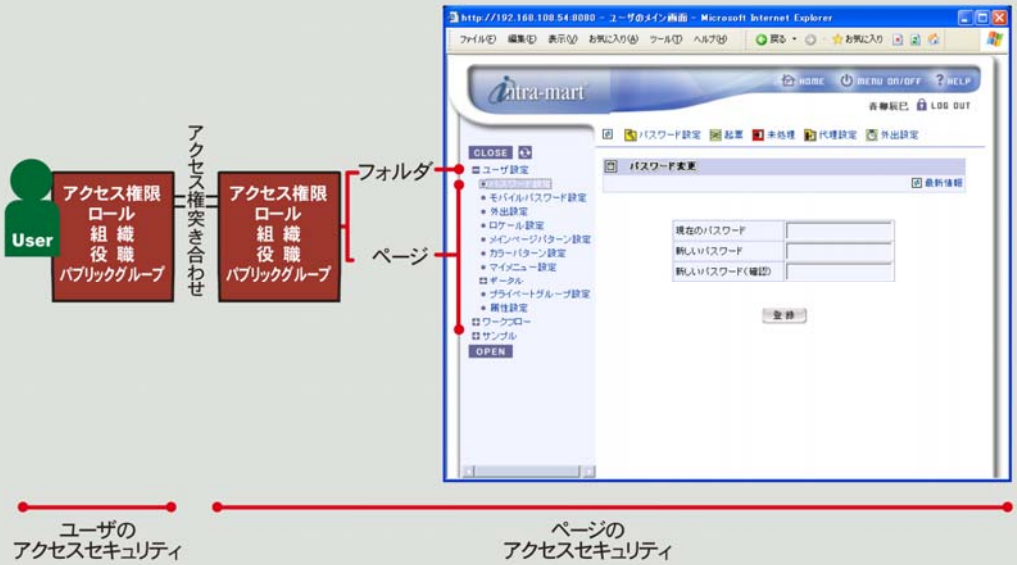
シングルサインオンを実現する アクセスセキュリティ・モジュール：IM-SecureSignOn	独自のエージェント型リバースプロシキ方式により適用範囲が広く、導入・運用が容易なシングルサインオンを実現するエクステンション・モジュールです。簡易ログイン認証機能により、他システムやパッケージとの連携も容易です。

- 詳細については、「システム管理者 操作ガイド」、「グループ管理者 操作ガイド」および「アクセスセキュリティ仕様書」を参照してください。

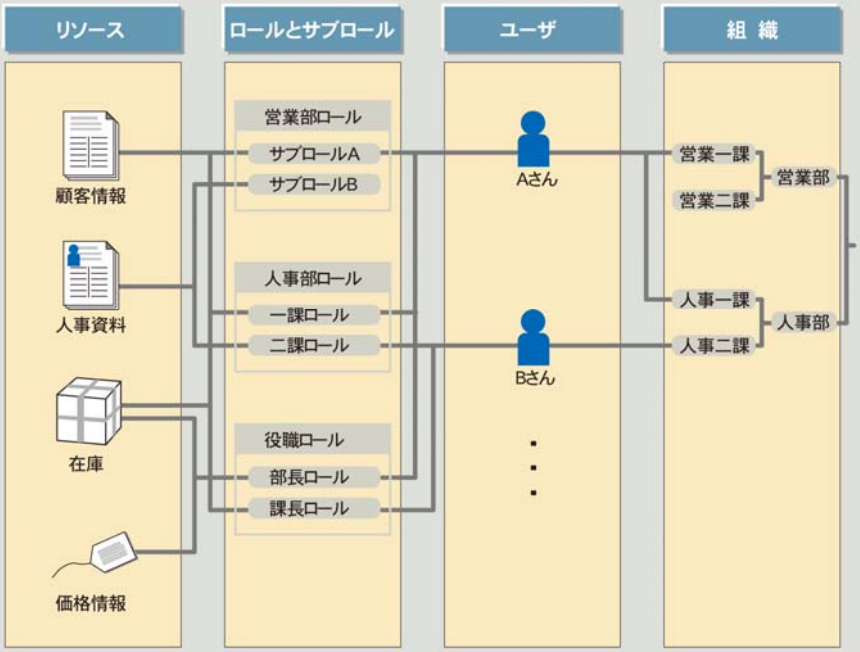
Column

intra-martのアクセスセキュリティの考え方

ｉntra-martのユーザには、ロール、組織、役職などの属性を設定することができます。そして、intra-mart画面のフォルダやページにも、ロール、組織、役職などを「アクセス権限」として設定することができます。ユーザがintra-martにログインすると、そのユーザが持つロール、組織、役職などの属性と一致するアクセス権限を持つフォルダやページだけが表示されるようになります。アクセスセキュリティのAPIが公開されています。外部ユーザアプリケーションから、intra-martのアクセスセキュリティの機能を利用することができ、企業内のアクセスセキュリティを一元管理することができます。



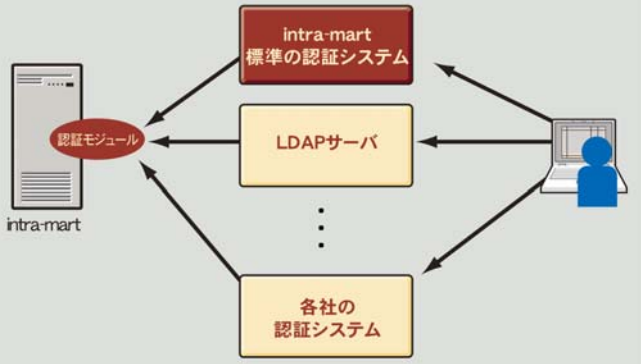
ロールによるアクセスセキュリティの設定例



Column

LDAPとの連携

intra-martは、ディレクトリサービスへの標準的なアクセス手段であるプロトコルLDAP（Lightweight Directory Access Protocol）にも対応しています。intra-martは、認証モジュールによって、intra-mart標準の認証システムをはじめ、LDAPサーバなどに接続して認証する形式となっています。この認証モジュールを変更することによって、各社の認証システムにも対応することができます。

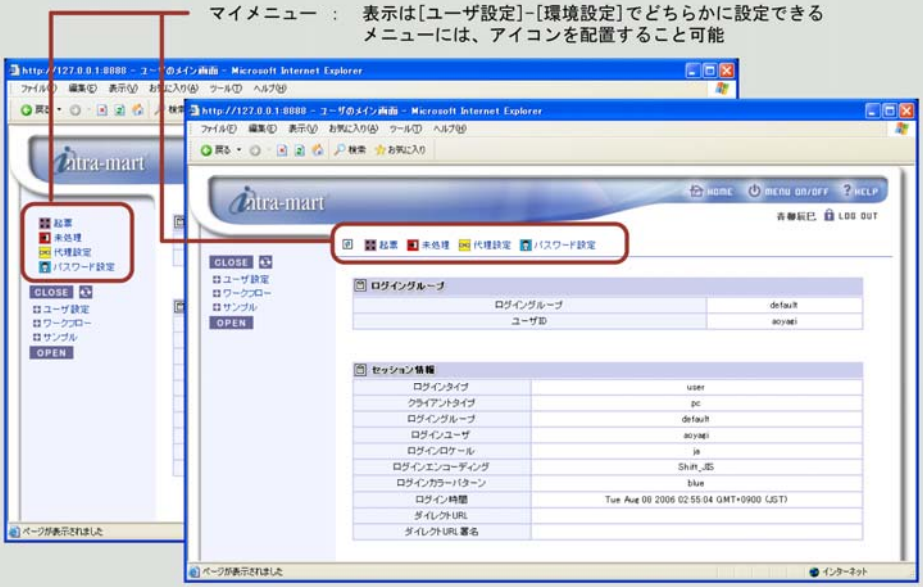


LDAPとの連携については「システム管理者 操作ガイド」 および 「アクセスセキュリティ仕様書」を参照してください。

Column

よく利用する画面を登録できる「マイメニュー設定」

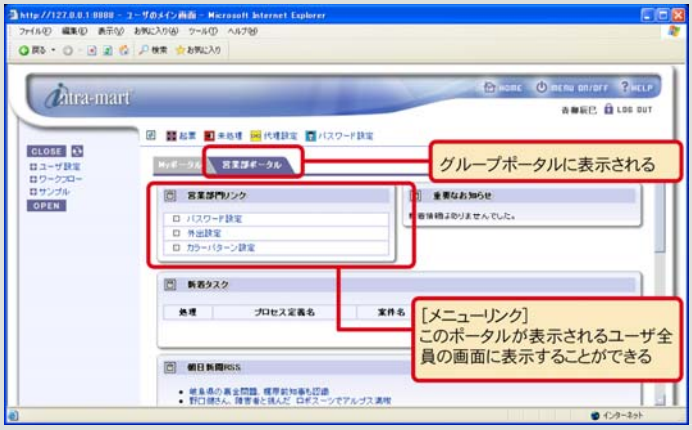
intra-martでは、頻繁に利用する画面を「マイメニュー」として登録することができます。登録されたメニューは、標準で表示されるメニューの最上部もしくは画面上部にアイコンとしてに表示され、メニューの階層をたどらなくてもワンタッチでアクセスできるようになります。「マイメニュー」へのページの登録は、［ユーザ設定］の［マイメニュー設定］メニューで行います。



Column

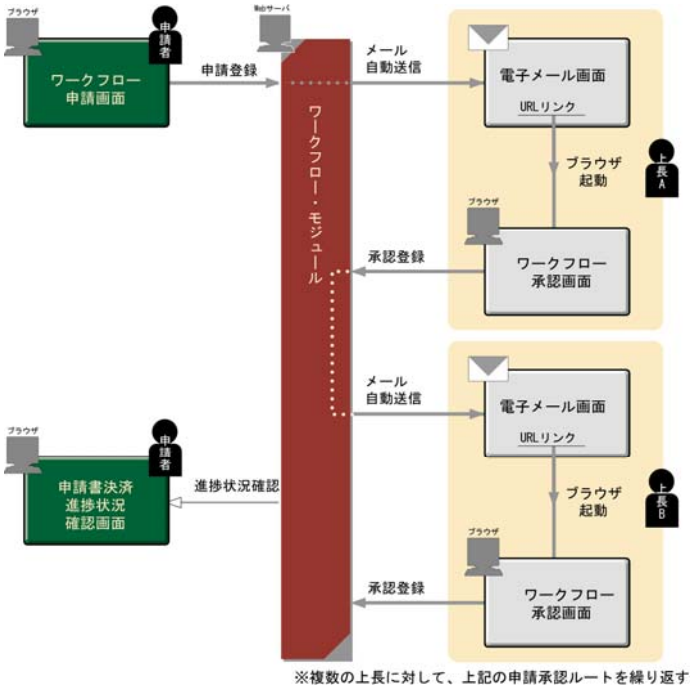
ポータル画面に表示する「メニューリンク」

メニューリンクはポートレットとして提供され、グループポータル画面上に任意のintra-martのメニューを表示させることができます。「マイメニュー」はユーザ個人ごとに設定するのに対して、「メニューリンク」はグループ管理者が設定し、指定したユーザ（アクセス権制御）のポータル画面に表示させることができます。



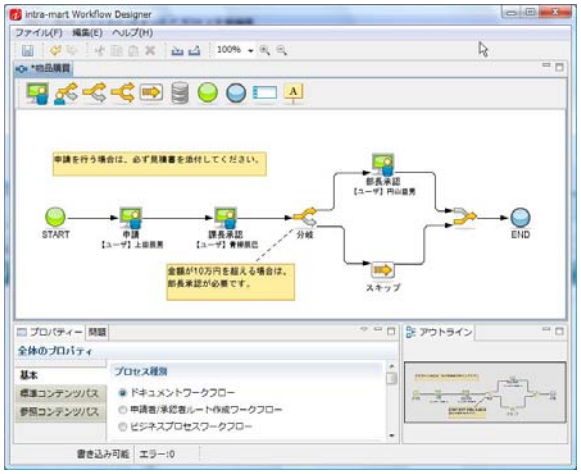
■ワークフローモジュール

Webブラウザベースのワークフローを構築するためのモジュールです。当モジュールを利用すると、イントラネット環境での多階層の承認ワークフローが構築できます。あらかじめ設定された承認ルートにしたがって、承認者宛てに承認ページURLリンクの入った電子メールが伝達されます。そこから承認者は自分の承認ページを起動し、承認行為を行います。さらに承認の進捗状況のチェックなど、きめ細かいワークフローの構築が可能になります。電子メールは、URLリンク機能およびSMTPに対応した市販のメールソフトに対応しています。
標準で用意されているワークフローモジュールのほかに、エクステンション・モジュールとして、「IM-EX申請システム（別売）」が用意されています。

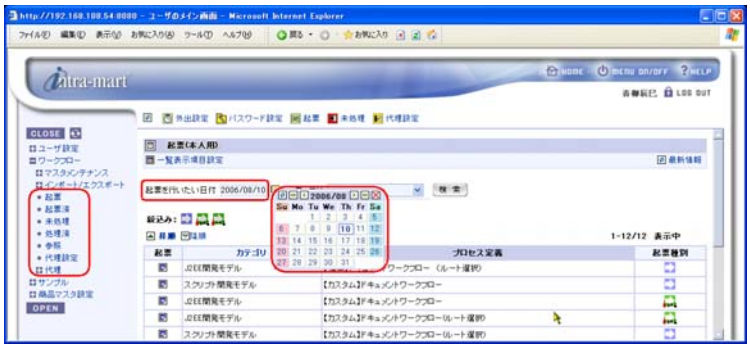


承認ルートはあらかじめ設定しておくことや、申請/承認時に次の承認者を画面から選択することができます。また、既存の人事システムと連動すると面倒な承認ルートの設定をなくすこともできます。
ワークフローモジュールの持つ高度な機能の例は以下のようになっています。

差戻し機能	案件を申請者または直前の承認者に戻す機能。
代理申請承認機能	申請者、承認者が不在時に代理者が申請承認を行う機能。
一括承認機能	承認者宛の複数の申請に対して、一括で承認/否認を行う機能。
根回し機能	現在依頼承認中であることを他のユーザにもメールで知らせる機能。
自動催促	承認期限がせまっている申請について、処置（承認/否認）を促す通知をメールで送信する機能。
自動パス機能	承認期限となったまたは承認期限を過ぎた申請について自動的に承認し、次の承認者に申請を送信する機能。
承認者の設定	承認ルート上の承認者は、個人だけでなく組織や役職でも指定可能（人事異動、組織改変に強い構造のルート設定が可能）。
引戻し機能	案件を引き戻しも可能。
承認案件振替機能	特定案件に対する承認権限を第三者に委譲する機能。
申請者/承認者ルート作成ワークフロー	稟議書の申請に代表されるような臨機応変なプロセスを実現するワークフロー。基本のプロセスは管理者が作成し、申請者／承認者が処理時に承認者を自由に選択することができます。



途中保存機能	申請前に登録内容を途中保存し、次回申請時に途中からスタートすることができる機能。
アプリケーション共通マスタの履歴化	アプリケーション共通マスタの履歴化に対応したルート定義ができる機能。将来変更予定の組織情報をもとに、ルート情報を作成することができる。
期間化されたルートの選択	期間化されたルートを過去・未来に渡って選択できる。異動が発生した場合にも、前職・現職どちらの立場で起票を行うかが選択可能。



ワークフローのステータスに対応したダイレクトメニューの採用	ワークフローのステータスをそのままメニューに採用しているため、直感的でわかりやすいメニュー。代理申請・承認も独立したメニューを用意（上図参照）。
きめ細かく設定できる代理設定	「プロセス定義単位」、「バージョン単位」、そして「タスク単位」、そしてこれらを組み合わせて、きめ細かく代理設定を行うことが可能。
項目の表示設定	起票、起票済、未処理、処理済、参照など一覧画面の表示項目を変更することができる。ワークフロー管理者があらかじめ表示項目を編集したセットを用意しておき、一般ユーザはこの中から任意に選択することが可能。
ワークフローモニタリング	完了案件ならびに未完了案件の状態をリアルタイムにモニタリングすることが可能。

- ワークフロー機能は「アドバンスト版」および「エンタープライズ版」のみの機能となります。
- ワークフローの詳細については、「ワークフロー操作ガイド」を参照してください。

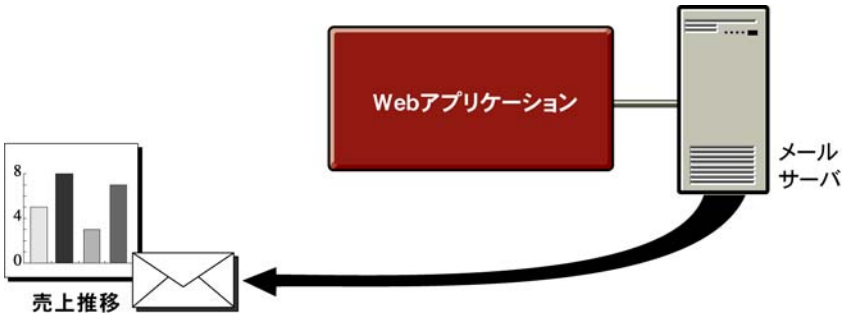
■バッチ管理モジュール

Webサーバ自身はもともとバッチ処理を想定していません。しかし、業務アプリケーションを構築する場合は、バッチで処理を実行したい時がでてきます。バッチ管理モジュールは、そうした要求を実現するためにサーバ上のサーバサイドスクリプトプログラムやJavaプログラムをバッチ実行できるスケジューラ機能を提供しています。intra-martの設定画面から、バッチプログラム（JavaScriptまたはJava）とプログラムの実行時刻を設定していきます。

● 詳細については、「グループ管理者 操作ガイド」を参照してください。

■メール連携モジュール

このモジュールを利用することで、他のSMTP互換のメールサーバに対するメール送信処理を行うなど、メールに連携したアプリケーションを簡単に構築することができます。たとえば掲示板にある情報を日付指定で指定ユーザに一斉にメール配信するエージェントを作成したり、売上情報を定期的に社員にメール配信するようなアプリケーションを構築したりすることができます。

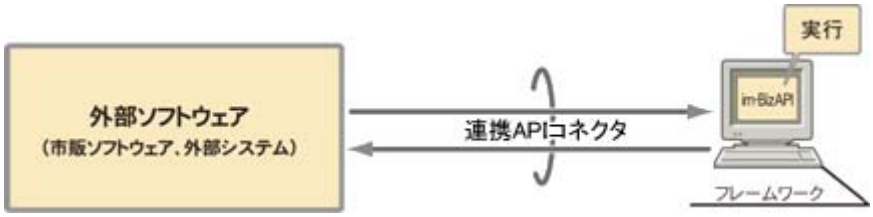


<売上情報などをメールで自動配信することができます>

● 詳細については、「スクリプト開発モデル プログラミングガイド」および「JavaEE開発モデル プログラミングガイド」を参照してください。

■外部ソフトウェア接続モジュール

intra-martアプリケーションと外部システムを連携するためのモジュールです。
このモジュールを利用することで、市販のソフトウェアや外部システムなどからリアルタイムに接続しim-BizAPIを呼び出して連携できます。

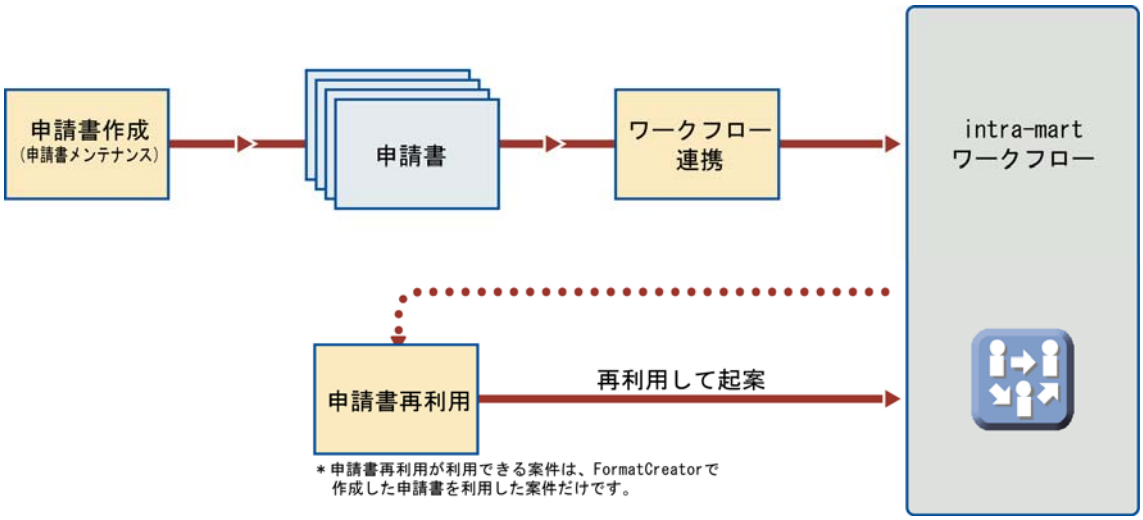


- 詳細については、「スクリプト開発モデル プログラミングガイド」および「JavaEE開発モデル プログラミングガイド」を参照してください。

3.1.3.3 業務基盤ツール

FormatCreator

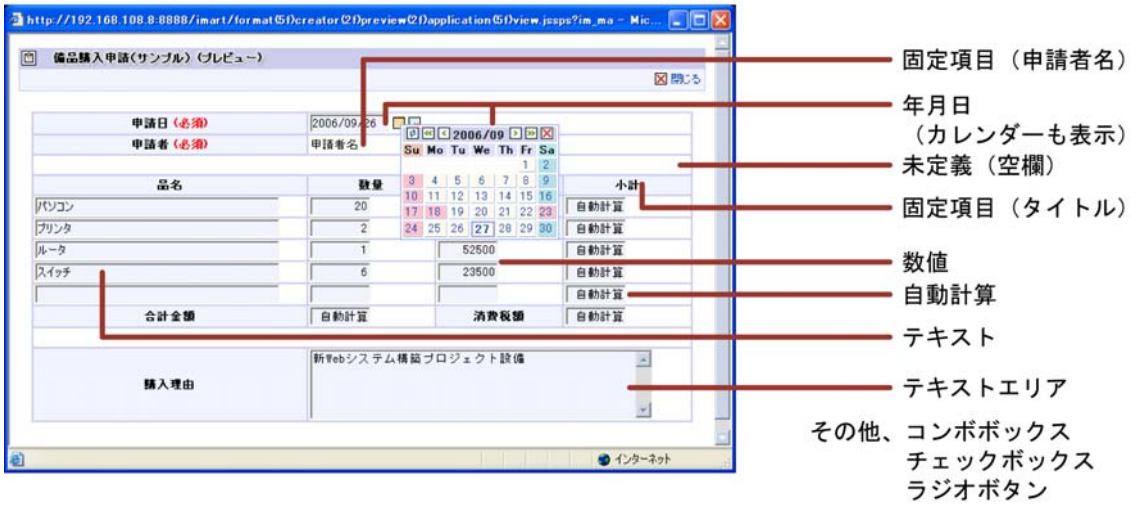
FormatCreatorは、intra-mart上で動作するドキュメントワークフローの起票画面（申請書）を、ノンプログラミングで作成するためのソリューションです。ウィザード形式の画面にしたがって設定するだけで、ワークフローの申請画面を作成することができるので、HTML、JavaScript、XMLなどのWeb画面作成言語、およびデータベースに関する知識を必要としません。申請書の入力項目に対する「入力必須」や「文字数」などの入力チェックも設定だけで機能するようになっています。
作成した申請書は、専用の連携画面で簡単にintra-martのワークフローと連携させることができます。
また、定期的に起票する案件に関しては、専用の起票画面から、以前に起票したものを再利用して効率よく起票することができます。



≪申請書項目≫

申請書の作成は、ウィザードにしたがって設定するだけで完成します。申請書に必要な項目は、次の図に示すような、さまざまな項目が用意されていて、それぞれに対して、入力チェックなどを制御する設定項目が用意されています。また、項目のレイアウトは、表作成のように、セルの結合と解除、行列の追加で行います。

作成した申請書の例（プレビュー）



≪メニュー構成≫

FormatCreatorには、次のような3つの機能が用意されています。

申請書メンテナンス	申請書の新規作成と編集を行う機能です。申請書は、新規に作成するほかに、既存申請書のコピーを作成して効率よく新しい申請書を作成する機能も用意されています。
ワークフロー連携	FormatCreatorで作成した申請書をintra-martのワークフローの起票画面に利用できるように連携させる機能です。連携には、ワークフローの新しいプロセスとするか、または既存プロセスの新しいバージョンとするかを選ぶことができます。
申請書再利用	FormatCreatorで作成した申請書を利用したプロセスを起票すると、申請書再利用の画面に一覧表示されるようになります。この画面では、この起票済みの案件を再利用して、効率よく新しく起票することができます。

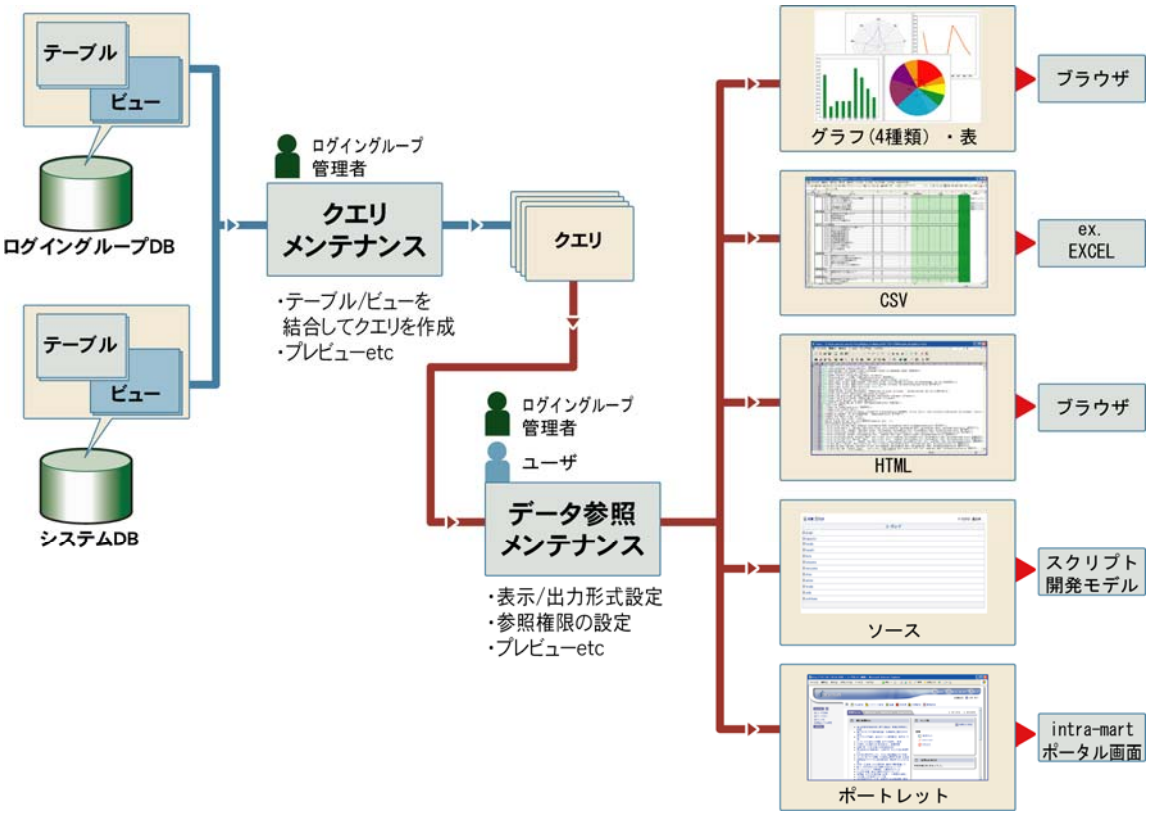
- 詳細については、「FormatCreator 操作ガイド」を参照してください。
- FormatCreatorは「アドバンスト」および「エンタープライズ for BPM」のみの機能となります。

■ ViewCreator

ViewCreatorはintra-martの画面上から、データベースのデータを使用して、様々な表やグラフを作成することができるツールです。使用可能なデータベースには、ログイングループデータベースとシステムデータベースがあります。ViewCreatorの機能は、クエリメンテナンスとデータ参照メンテナンスの2つの操作に大きく分かれます。

クエリメンテナンス	データベース上のテーブルやビューの結合などを行い、データ参照で使用する元データとなる表を作成します。また、作成したクエリのSQL表示やViewの作成、あるいは表をプレビュー表示することもできます。
データ参照メンテナンス	クエリメンテナンスで作成したデータの表示の仕方（グラフまたは表など）やデータの絞込みに関する設定を行います。また、参照権限をデータ参照単位で設定することができます。

作成したデータ参照は、表示時にデータの検索や表示項目の絞り込み、並び順の変更などを行うことができます。また、データ参照はポートレットとして追加したり、CSV/HTML形式のファイルとして出力、あるいはスクリプト開発モデルのプログラムファイルとして出力可能で、出力されたプログラムソースは自由にカスタマイズして再利用が可能です。
このように、データベースの中のデータを元に様々な表やグラフの作成/表示をWebブラウザ上で簡単に操作できるのがViewCreatorの大きな特徴です。



<ViewCreator>

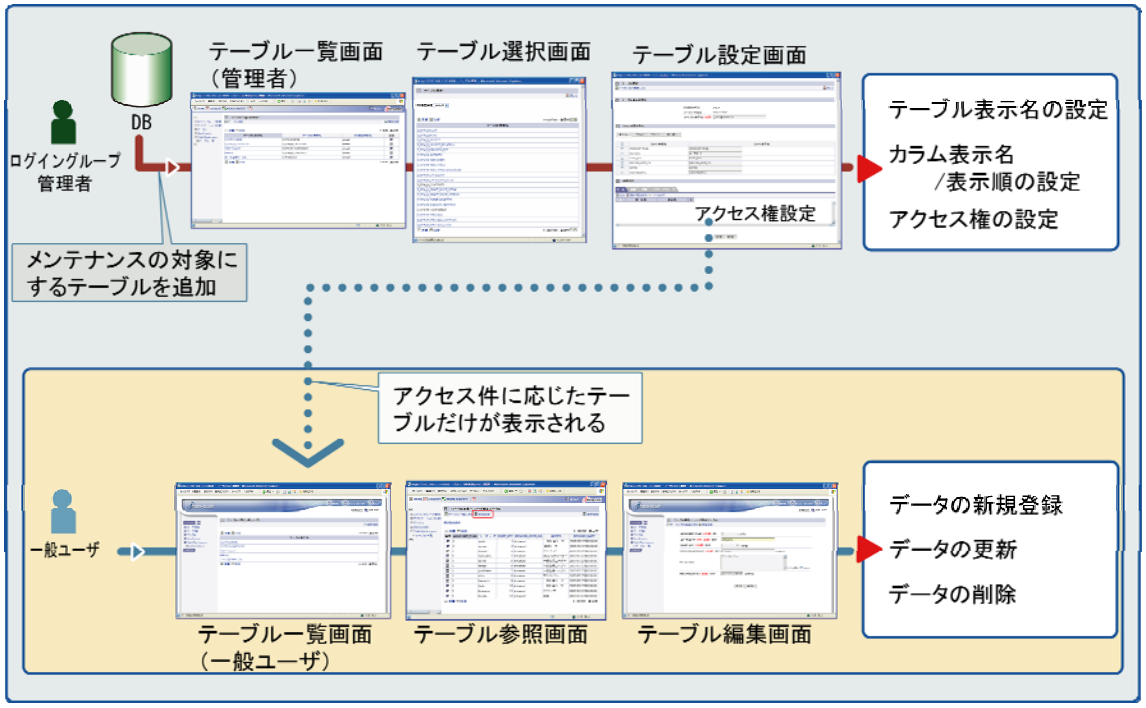
- 詳細については、「ViewCreator 管理者操作ガイド」および「ViewCreator ユーザ者操作ガイド」を参照してください。

TableMaintenance

TableMaintenanceでは、データベース上の既存のテーブルに対してレコードの新規登録・更新・削除を行うことができます。データベースを使用したアプリケーションを作成した際に必要になることの多いマスタメンテナンスの機能を簡単な設定操作のみで、簡易的に実現することができます。ログイングループ管理者が、メンテナンスを行うテーブルに対してアクセス権の設定を行うことで一般ユーザが参照できるテーブルを制限することもできます。TableMaintenanceで扱えるデータベースは、ログイングループデータベースとシステムデータベースです。

TableMaintenanceの操作は、以下の2つに分類することができます。

テーブルの管理	データベース上のテーブルをメンテナンスの対象とするテーブルとして追加します。また、アクセス権を設定して、そのテーブルを操作できるユーザを限定することもできます。この操作は、「ログイングループ管理者」が行います。
データの編集	管理者からアクセス権を与えられたユーザは、テーブルのデータの追加・更新・削除が行えるようになります。

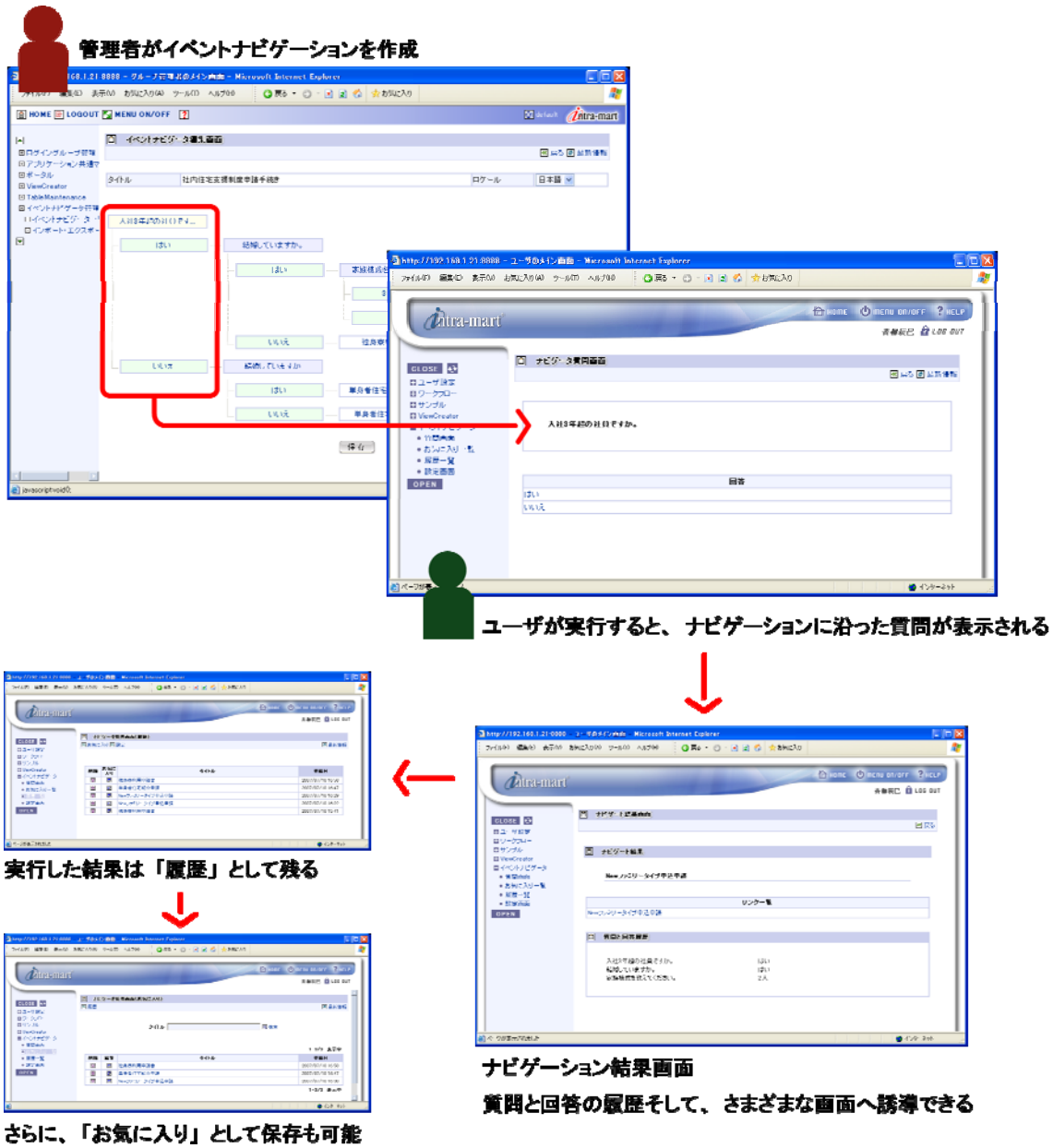


< TableMaintenanceの概念>

- 詳細については、「TableMaintenance 管理者操作ガイド」を参照してください。

■イベントナビゲータ

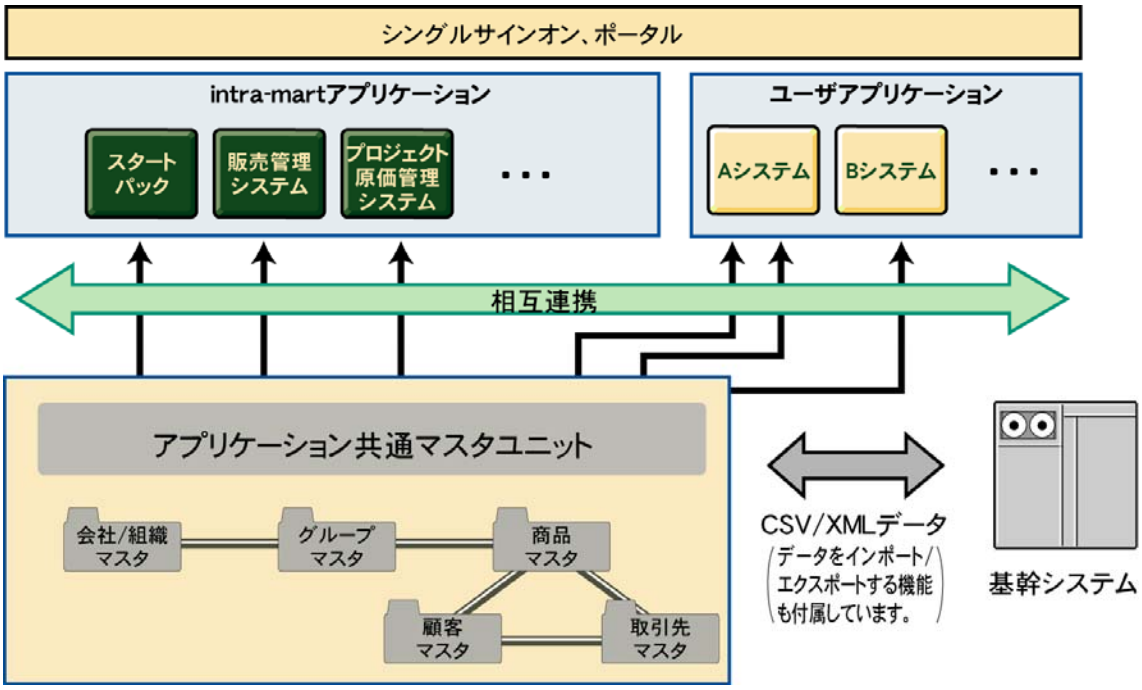
イベントナビゲータは、条件によって複雑に分岐する申請書などの画面へのアプローチを、ウィザード形式で導く画面を作成するメニューです。
下記の例は、会社内の社宅規定にもとづき、適切な申請を導くサンプルです。表示される質問に、答えてゆくだけで、最適な申請書にたどりつけるようになります。



<イベントナビゲータの概要>

3.1.3.4 アプリケーション共通マスタ

会社データ、組織データ、グループデータ、取引先データ、顧客データ、商品データなど、システム開発でよく利用するマスタが標準で用意されています。これらのマスタを利用することにより、設計工程まで含め短期間でのシステム開発が可能となります。また、各intra-martアプリケーションシリーズは、すべてこのアプリケーション共通マスタをベースに構築されていますので、各intra-martアプリケーションシリーズと連携したシステムが開発できます。さらに、マスタにアクセスするためのAPIなども標準で用意されています。



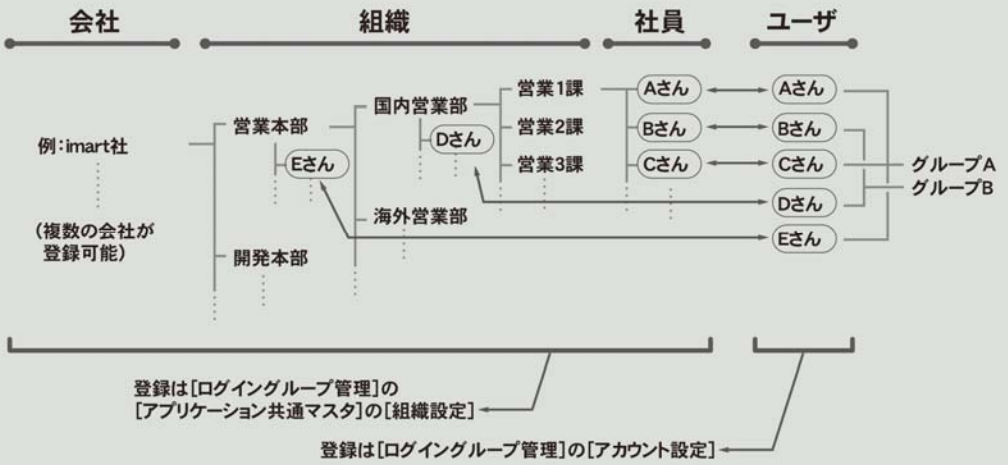
これらのマスタはすべてデータベース仕様も公開されており、さらに用意されているAPIを利用しながらユーザアプリケーションを作りこんでいくことができます。また、各intra-martアプリケーションシリーズも、アプリケーション共通マスタを利用して構築されていますので、必要なデータを相互連携させながらアプリケーションを拡張していくこともできます。

- 詳細については、「アプリケーション共通マスタ仕様書」、「アプリケーション共通マスタAPIガイドライン」を参照してください。

Column

アプリケーション共通マスタで定義される組織とは

「組織」とは、会社ー組織（事業部・部・課など）ーユーザというデータの階層を指します。会社を登録し、その会社に属する組織として、事業部や部・課・係などの組織を登録します。組織を階層化したり、組織のどの階層に対しても、ユーザを紐づけたりすることができます。



※ 上記図中の「グループA」と「グループB」がパブリックグループに相当します。

Column

アプリケーション共通マスタで定義されるパブリックグループとは

パブリックグループは、企業内の部門など組織にとらわれないユーザグループを作成したい場合に利用します。

Column

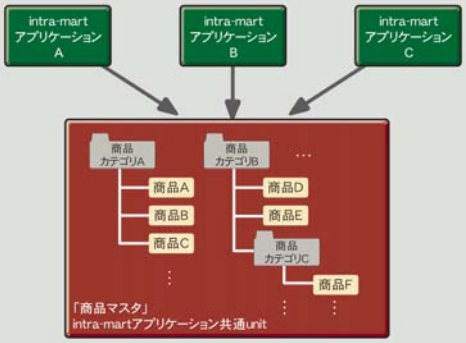
アプリケーション共通マスタで定義されるプライベートグループとは

パブリックグループと同様に、組織に関係なく自由に作成できるユーザグループで、ユーザ自身で自由にユーザグループを作成できるように用意しています。

Column

アプリケーション共通マスタで定義される商品マスタとは

intra-martで作成する複数のアプリケーションから共同で利用することができる商品マスタが標準で用意されています。



Column

アプリケーション共通マスタの国際化対応

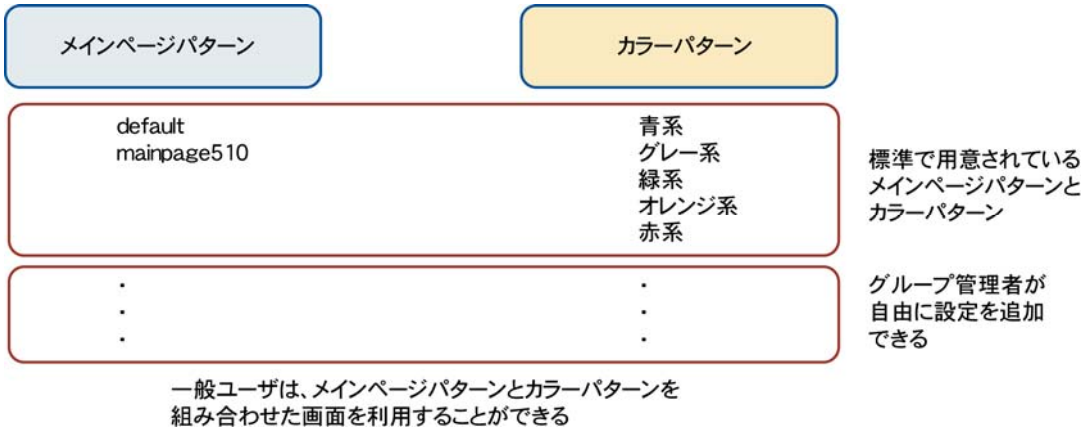
アプリケーション共通マスタは、国際化対応されています。例えば会社名や組織名などを多言語登録することができます。また、登録したデータはロケールを指定して検索することにより、その時々に応じて必要な言語でデータを扱うことができます。さらに、アクセスセキュリティのAPIを利用してログインセッションのロケールと連動させることによって、アプリケーションを国際化対応させることができます。

3.1.4 標準画面の作り方（共通画面デザイン）

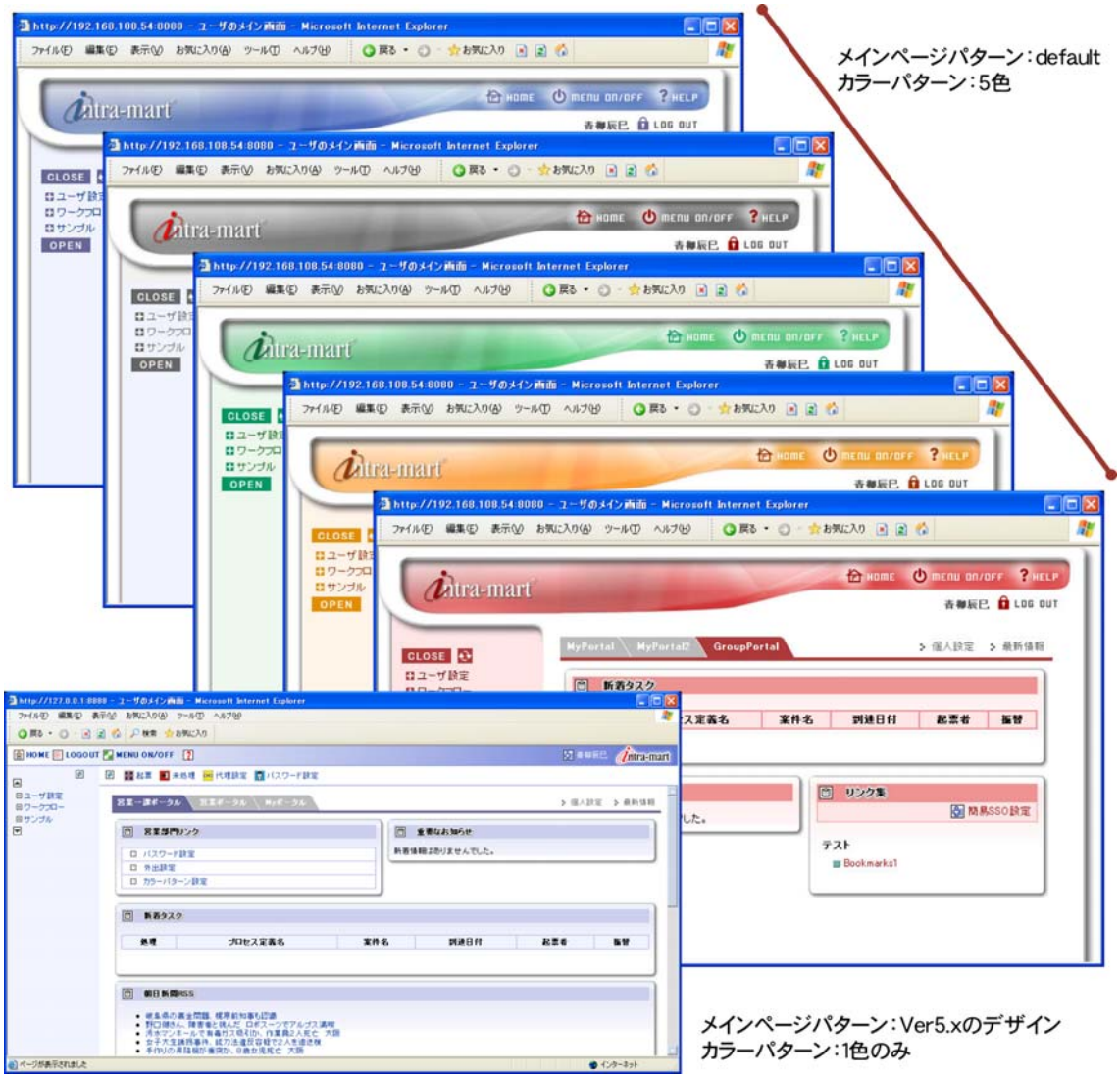
intra-martの標準画面は、提供されている共通化ＡＰＩを利用して、オリジナルな画面デザインでアプリケーションを開発することができます。画面のデザインは、「メインページパターン」と「カラーパターン」の２つから構成されています。メインページパターンは、HTMLまたはJSPなどで作成された画面のデザインページで、カラーパターンはCSSファイルとなっています。これらのファイルは、グループ管理者が用意します。一般ユーザの画面には、グループ管理者が用意した、「メインページパターン」と「カラーパターン」が表示され、その組み合わせの数だけ画面をカスタマイズすることができます。
標準の画面には、２つのメインページパターンと５つのカラーパターンが用意されています。
デザインが共通化されると、他の画面と見た目や操作性が統一され、メニューから呼び出されたときに利用者が違和感なくアプリケーションを操作できるなどのメリットがありますので、アプリケーション開発の際には、デザイン統一の方法としてこのガイドラインの利用を検討してください。画面作成に関しては、以下のドキュメントが用意されています。

- 画面デザインガイドライン
- 画面デザインスタイルシート仕様書
- アクセスセキュリティ仕様書

画面デザインガイドラインに準じたＡＰＩも用意されています。



- API仕様は、「画面デザインガイドライン」、または「APIリスト」を参照してください。
- WebPlatform/AppFrameworkの持つ各画面ソースは、そのほとんどがプレーンな状態でインストールされています。画面デザインガイドラインの適用方法や、APIの使用例としてご活用ください。



<標準で、2つのメインパターンと5つのカラーパターンが用意されている>

Column

intra-mart WebPlatform/AppFrameworkを利用する上での規定事項

intra-mart WebPlatformならびにAppFrameworkを利用したアプリケーション構築には、いくつかの規定事項があります。

1　ユーザ設定画面の利用
intra-martへのログインユーザ管理は、「ユーザ設定」画面を使用しています。アプリケーション構築の際にも、ログインユーザの管理は、そのまま「ユーザ設定」を利用してください。アクセスセキュリティ・モジュールやワークフロー・モジュールなどは、このユーザ設定で作成されたアカウント情報の利用が前提で稼動します（次のColumnを参照してください）。
ユーザ管理マスタの登録については「グループ管理者 操作ガイド」を参照してください。

2　フォルダ/ページのメニュー構成
intra-martのメニュー構成は、メニューツリーのフレーム利用が前提となっています。特に、アクセスセキュリティ・モジュールは、このメニュー構成に従うことが前提となっています。フォルダやページへの登録については、「グループ管理者 操作ガイド」を参照してください。ただし、アクセスセキュリティの各機能はAPI化されており、それらを利用して独自のメニュー画面を作成することができます。
また、これらのログインユーザ管理機能やフォルダ/ページ機能を利用せずにintra-martをDBと連携した機能を利用することも可能です。

Column

intra-martで提供する各モジュール(im-BizAPI)のアカウント情報への依存について

intra-martの各モジュールには、intra-martのアカウント情報（「ユーザ設定」画面でのユーザ管理）の利用時にしか使用できないものもあります。下記表を参考にしてください。また一部モジュールには、別のモジュールの利用を前提として使用可能となるものもあります。下記の注意点を参考にしてください。

intra-martモジュール(im-BizAPI)	アカウント情報なしで： ○利用可能、△一部利用不能、×利用不能
画面共通モジュール	△＊3
ポータルモジュール＊1	×
グラフ描画モジュール	○
カレンダー表示モジュール	○
ツリー表示モジュール	○
アプリケーション共通モジュール	△
メール連携モジュール	○
ワークフローモジュール＊1	×
アクセスセキュリティモジュール	×
バッチ管理モジュール	○
アプリケーション共通マスタ	×
エクステンションモジュール	
IM-PDFデザイナー	○

＊1　アクセスセキュリティモジュールの利用が前提のモジュールです（intra-martのアカウント情報をプログラム内部で使用して動作するモジュールです）。
＊3　デザインタグはアカウント情報に依存しています。

- アカウント情報の仕様については「アクセスセキュリティ仕様書」を参考してください。

3.1.5 エクステンションシリーズ

エクステンションシリーズは、intra-mart WebPlatformやintra-mart AppFrameworkに標準添付されているモジュールとは別に用意されている拡張モジュール群（オプション製品）です。より高機能なモジュールが必要なユーザのために用意されており、必要に応じて組み込んで標準のモジュールと同様に利用することができます。現在用意されているモジュールには、次のようなものがあります。

3.1.5.1 IM-PDFデザイナー（別売）

PDFを採用することにより、より複雑な帳票形式やページ制御の対応が可能となるモジュールで、単票および連票に対応します。ビジュアルな帳票デザイン作成ツール「IOWebDoc」で帳票フォーマットを作成します。ユーザアプリケーションからのデータはCSV形式またはメモリ経由で受け渡し、PDFファイルを作成してからAcrobatを起動して印刷します。複数のアプリケーションサーバを配置した環境では、IM-PDFデザイナーをStorage Service上で稼動させることによって生成したPDFファイルを一元管理して、クライアントへダウンロードすることが可能になります。

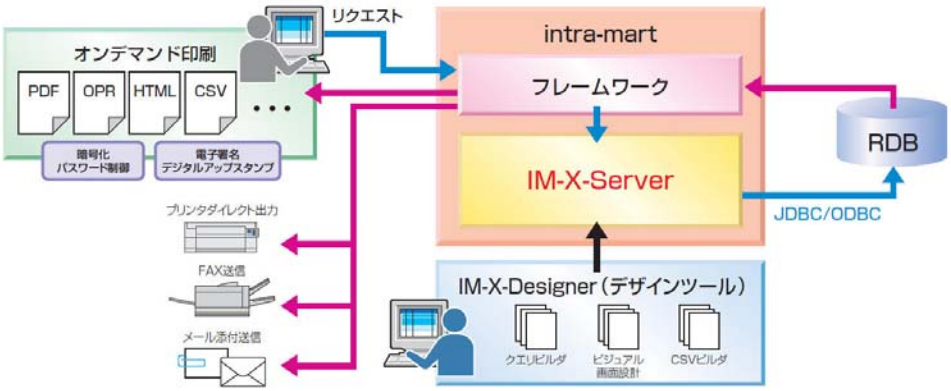
<IM-PDFデザイナーサンプル画面>

3.1.5.2 IM-X Server(別売)

大量帳票出力やプリンタへのダイレクト出力、電子署名とタイムスタンプのサポートも可能な印刷モジュールです。XML対応の高機能ビジネス帳票ソリューションで、オンデマンド印刷、ダイレクト出力など、多様な機能を提供します。

≪さまざまな形式の帳票の生成・出力・配信が可能≫

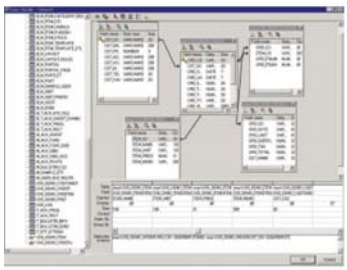
IM-X-Serverの帳票生成はXMLで定義されるため、電子化（PDF、HTML、CSV、OPR）や印刷（ダイレクト印刷、FAX送信）から検索・入力フォームまで、１ソースマルチフォーマットで生成でき、開発工数を削減することが可能です。基幹帳票生成から日々の業務に必要なビジネスレポートまで、トータルソリューションを提供します。



≪きめ細かい帳票作成が可能なデザインツールも合わせて提供≫

日本固有の複雑な罫線やページ単位のレポート、連帳・単票・サブレポート、ラベル、カスタムサイズ等の設計に対応。動的グラフ生成、動的バーコード生成、電子署名、表計算、複数オブジェクトのグループ化による段組表現など、フレキシブルな表現でさまざまなビジネス帳票に対応します。

[画面設計]
GUI部品の配置、プロパティ入力で直感的操作が可能。
各種ウィザードの利用で設定不要

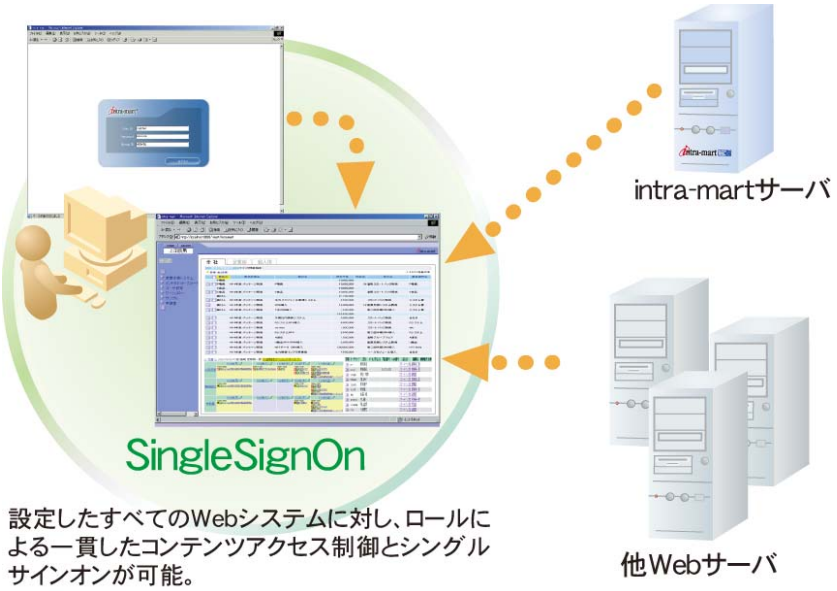


[クエリビルダ]
データベース種類を意識せずに設定が可能。
GUIでDB、CSV、XMLのデータを取得。

[アクセス制御にも対応]
PDFサンプル/ドキュメントセキュリティ・電子署名対応。
オンデマンドパスワード。

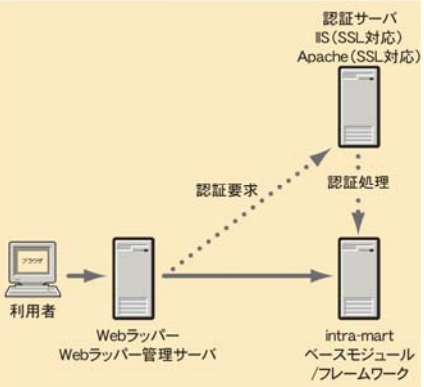
3.1.5.3 IM-SecureSignOn（セキュア・サイン・オン）（別売）

IM-SecureSignOnは、シングルサインオンを実現するツールです。社内のさまざまなWebシステムへのログインも、このSecureSignOnに一度ログインするだけ、すべての認証が完了します。独自のエージェント型リバースプロキシ方式＊により、適応範囲が広く、導入・運用が容易なシステムです。また、簡易ログイン認証機能により、他システムやパッケージとの連携も容易です。



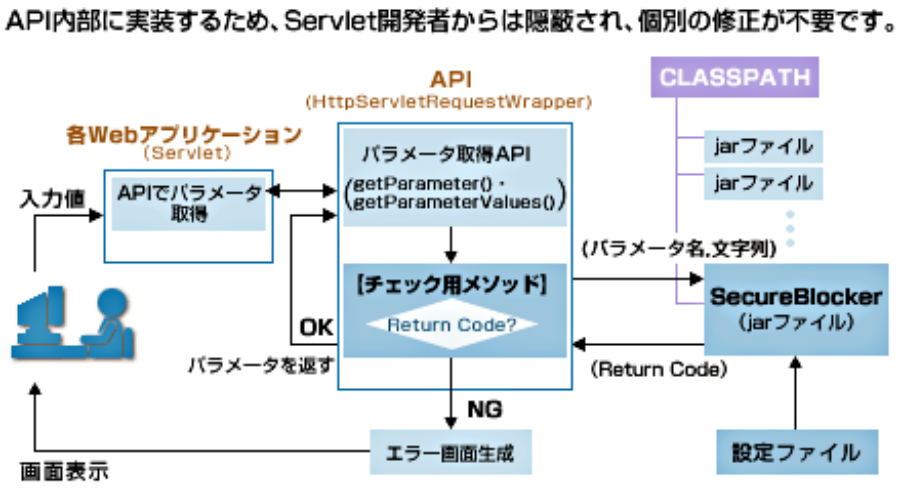
≪エージェント型リバースプロキシ方式≫

シングルサインオンを実現するには、一般的にリバースプロキシ方式とエージェントモジュール方式の2つの方式が考えられます。IM-SecureSignOnでは、これら両者の長所を兼ね備えた独自のエージェント型リバースプロキシ方式を採用しています。

方式名	主な用途	特 長	システムの概念図
エージェント型リバースプロキシー方式	管理分散型イントラネットに適したシングルサインオンの方式	・負荷が集中しない ・各エージェントが直接アクセスを受ける ・大規模ユーザに対応可能 ・Webサーバの分散配置構成に対応可能 ・Webサーバ、ホストOSの種類を問わない	

3.1.5.4 IM-SecureBlocker(別売)

インターネットに公開するWebアプリケーションのセキュリティ対策を低コストで実現することができます。脆弱な箇所にピンポイントで適用が可能なため、導入によるWebアプリケーション全体への影響を最小限に抑えることができます。
SecureBlockerは入力パラメータを検査し、自動的に無害化してくれるJavaクラスライブラリです。

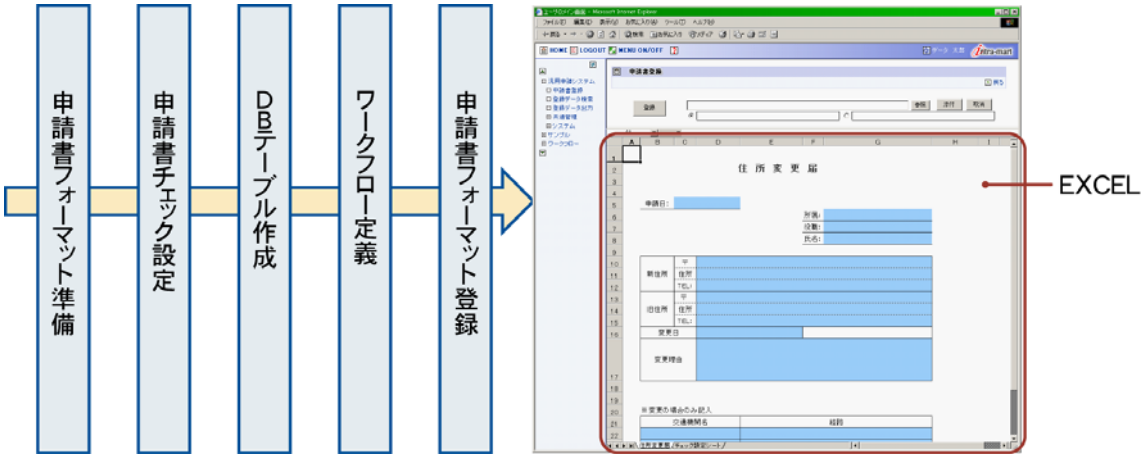


IM-SecureBlockerを導入すると、次のような効果が得られます。

- HTTP Request のパラメータの入力値チェック機能の検討・実装コストが削減できます。
- 開発者のスキルに依存せず、Web アプリケーション脆弱性対策が実施できます。
- Web サイトで使用するパラメータごとにチェックを制御可能です。
- クロスサイトスクリプティング、OS コマンドインジェクション、ディレクトリトラバーサル、SQL インジェクションのチェックが可能です。

3.1.5.5 IM-EX申請システム（別売）

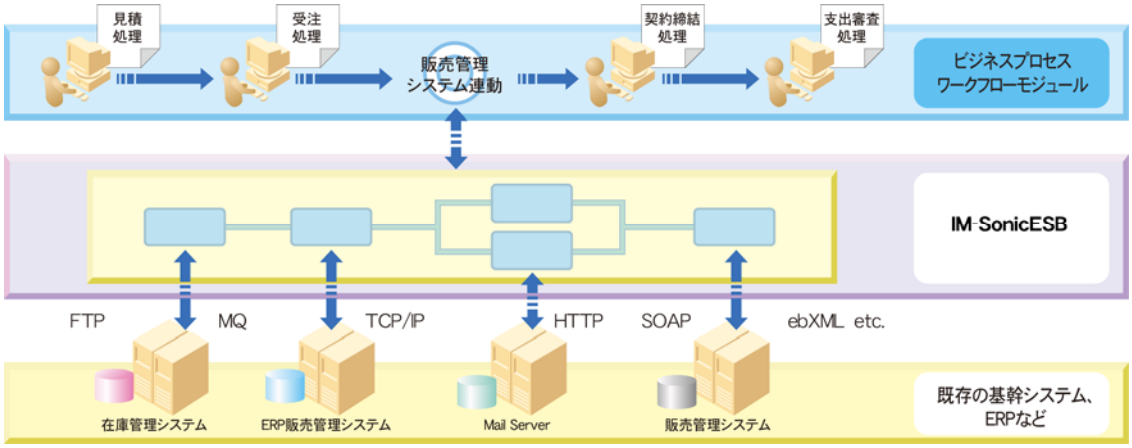
電子申請のための申請書フォーマット画面にExcelシートを利用することができるソリューションです。intra-martのワークフローと組み合わせて利用することで、申請ワークフローを設定だけでノンプラグラミングで作成することができます。書式設定やマクロはExcelの機能をそのまま利用することができるので、入力チェックを行うこともできます。入力した情報はデータベースの指定のテーブルに保存することが可能です。



- Excel様式さえ用意すれば、短時間で申請書をWeb化することができます。
- お使いのExcelの様式を入力画面に活用し intra-mart のワークフロー連動することができます。
- Excelの書式設定や提供するマクロを設定し、クライアントサイドで入力チェックが行えます。
- Excelに入力した情報は、データベースの指定のテーブルに保存可能です。
- データベースに保存されている情報を指定のExcel様式で出力できます。
- Excel様式の登録、入力チェック等、すべて設定だけでプログラム言語の知識がいりません。

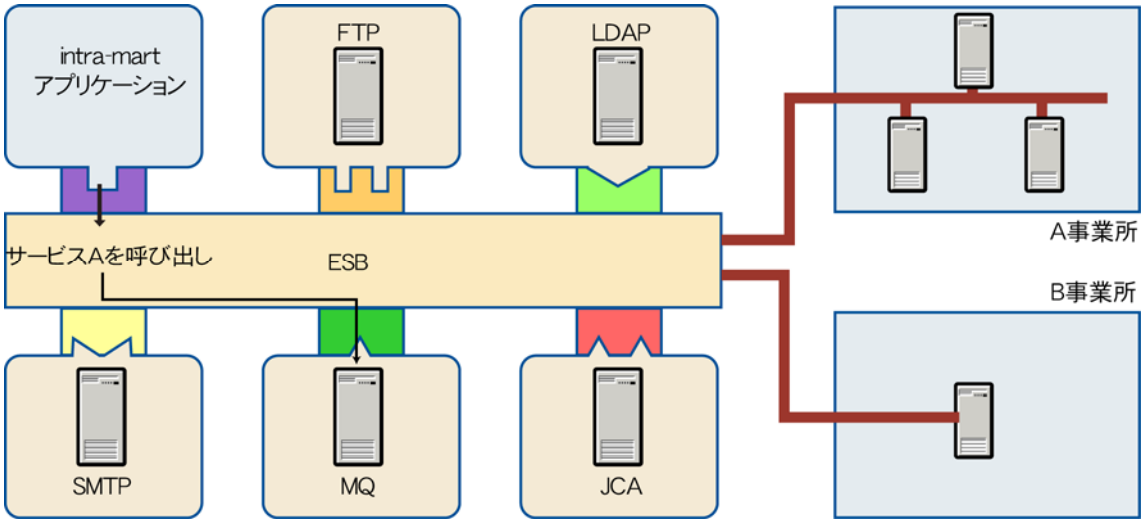
3.1.5.6 IM-SonicESB(別売)

IM-SonicESBは企業内に分散している複数の既存システムを連携する、バックエンドシステム統合プラットフォーム(ESB)です。システム間のデータの一貫性を保証する機能も実装しており、Webサービス同士を高信頼に連携させるアプリケーションを、容易に開発することが可能です。
ビジネスプロセスワークフローモジュールと連携することで、バックエンドも含めたダイナミックかつミッションクリティカルなシステム統合が実現します。



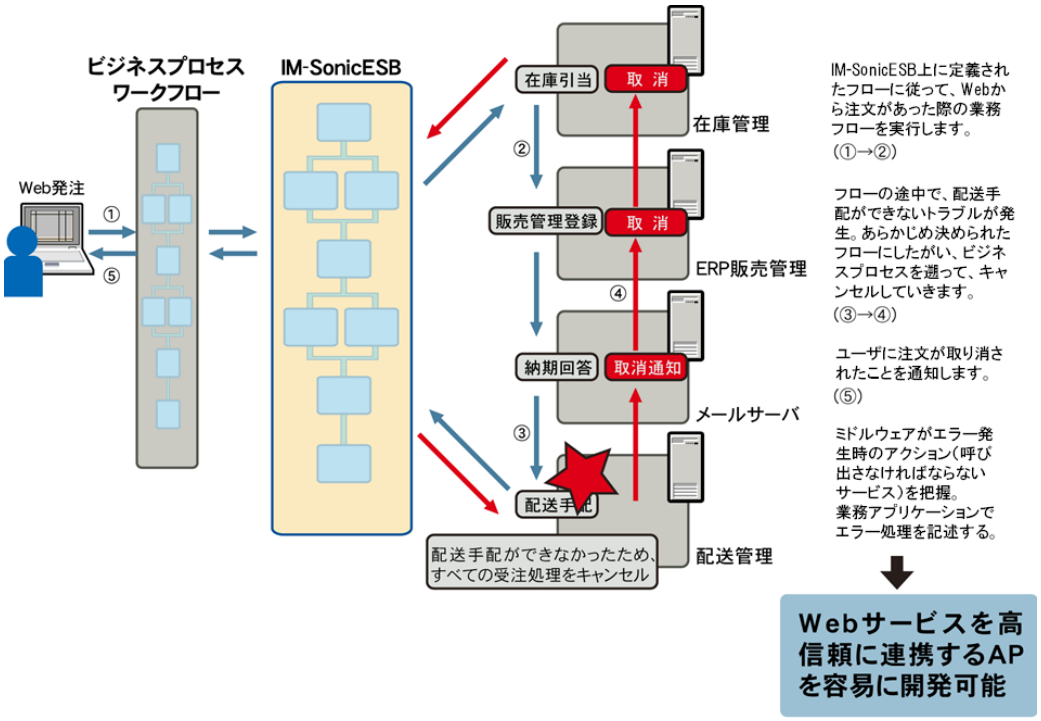
≪ショート&ロングランニングプロセスに対応したシステムを実現≫

intra-martのビジネスプロセスワークフローは、さまざまなWebシステムを業務フローに従って、柔軟に連携させることができます。いわゆるロングランニングプロセスを実現しています。一方、IM-SonicESBは、企業内のさまざまな既存システムを高信頼で結ぶ「ショートランニングプロセス」を実現しています。
そして、intra-martのビジネスプロセスワークフローからIM-SonicESBと連携することにより、複数の業務プロセスや業務システムを統合・制御・自動化し、業務フロー全体を最適化するビジネスプロセスマネジメント（BPM）が実現します。



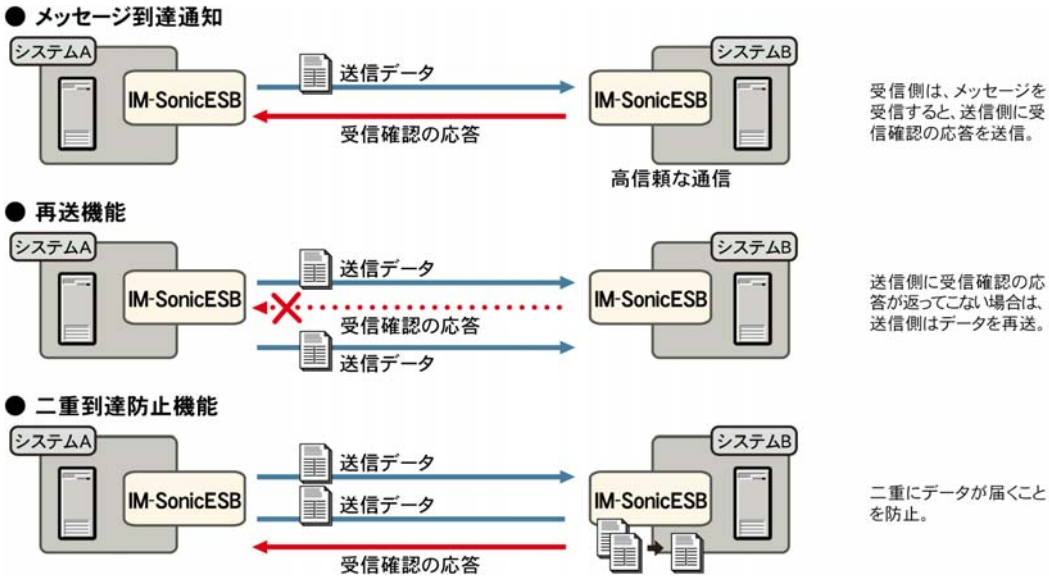
≪IM-SonicESBの高信頼性 ― トランザクション≫

IM-SonicESBは、エラー発生時に連携システム間のデータをの一貫性を保証する機能を持っています。



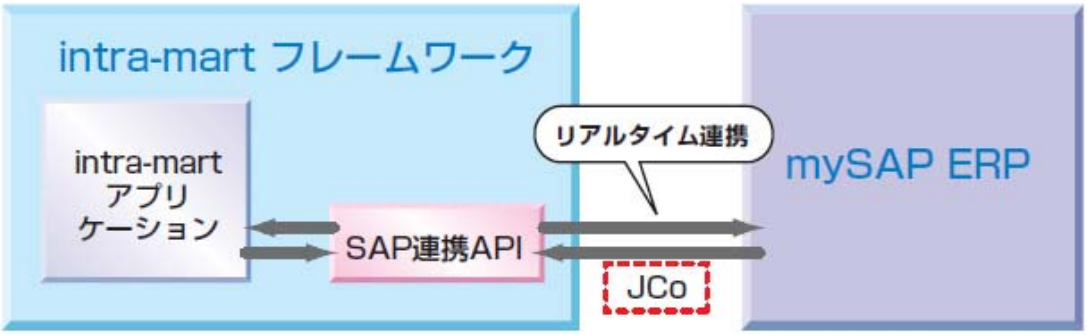
≪IM-SonicESBの高信頼性 ― メッセージ保証≫

IM-SonicESBは、HTTP上の高信頼なメッセージを実現するためのさまざまな機能を持っています。



3.1.5.7 IM-ERPリアルコネクト（別売）

intra-martでは、ERPとリアルタイムに連携できるエクステンションモジュールを用意してます。
SAP JCo技術を利用してSAP 連携APIをライブラリ化しています。標準のJava技術を用いて、アドオン開発を低コストで行うことが可能になります。従来のバッチを中心にした連携から、リアルタイムトランザクション連携へとWebの可能性が大きく拡大されます。

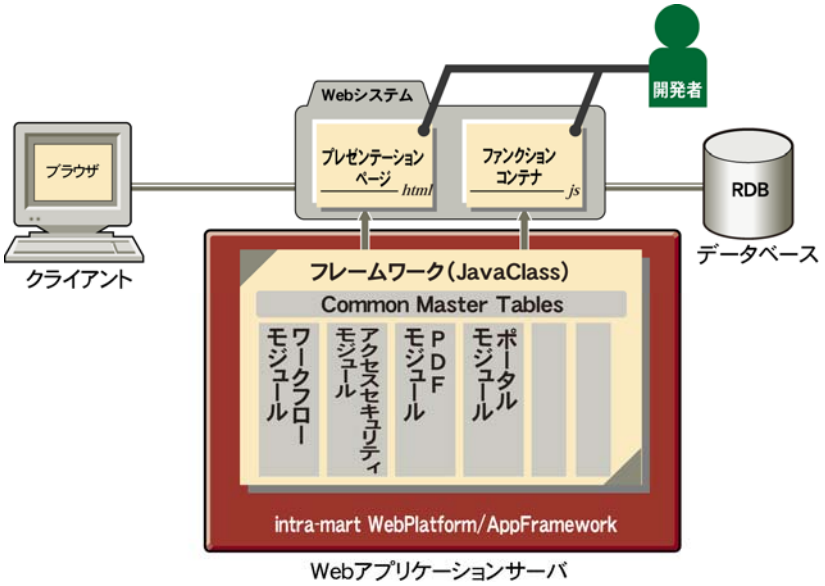


- 現在提供しているIM-ERPリアルコネクトには、SAPとの連携ができる豊富なAPIが用意されています。その他のERP用の連携モジュールも順次追加して行く予定です。
- 詳細については以下の資料を参照してください。
  - 製品に添付されている、SAP R/3 および ERP6.0とリアルタイムに連携するためのプログラミングガイド「SAP R/3 連携 プログラミングガイド」「SAP ERP6.0連携 プログラミングガイド」
- SAP、mySAP ERP、SAP R/3、SAP ERP6.0、SAP JCO、製品内に記載するSAPの製品/サービス名は、すべてドイツおよびその他の国におけるSAP AGの商標または登録商標です。
- 2004年6月にSAP Certified Integration及びPowered by SAP NetWeaverTMの認定を取得しました。

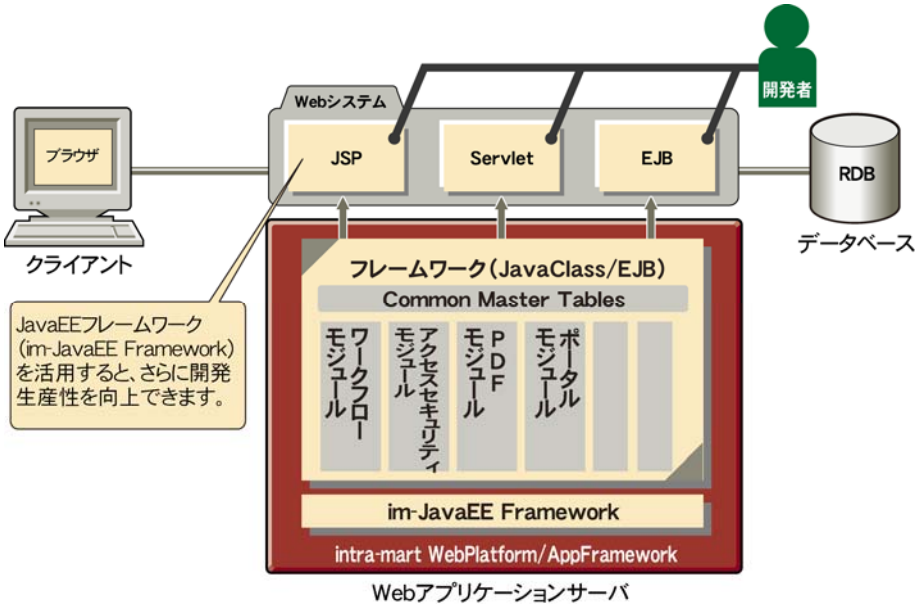
3.2 intra-martのアプリケーション開発概要

intra-mart WebPlatformあるいはintra-mart AppFrameworkを使ったアプリケーションの開発において、開発者はブラウザ上に表示されたユーザインタフェースと、Webサーバ上で動作するビジネスロジックを作成することになります。

スクリプト開発モデルではプレゼンテーション・ページ（HTMLファイル）とファンクションコンテナ（サーバサイドJavaScriptファイル）の2つのファイルを作成します。この際、フレームワークで用意されているモジュール群（im-BizAPI）を活用することでさらに生産性を向上させることができます。



JavaEE開発モデルでは、JSPファイルとServlet、EJBコンポーネントで開発します。
この際にも、フレームワークで用意されているモジュール群（im-BizAPI）をJavaEEフレームワーク（im-JavaEE Framework）とともに利用することで、煩雑なJavaEEによるWebシステム開発をさらに効率化するとともに、業務コンポーネントの再利用を促進することができます。



Column

2つの開発モデルの使い分け

JavaEE開発モデル
コンポーネント再利用や並行分散開発により、大規模システム開発において開発生産性を発揮します。

スクリプト開発モデル
コンポーネントアーキテクチャではなく、1ファイルに業務処理を記述していくスタイルであるため、少人数システム開発において生産性は高くなります。

※　ただし大規模開発においても、コンポーネント再利用性の低い画面（マスタメンテナンスなど）においては軽量プログラミングを使用するなど、双方を組み合わせることでコスト削減が可能
（すべてをJavaEEで開発することはオーバーヘッドが大きくなります）

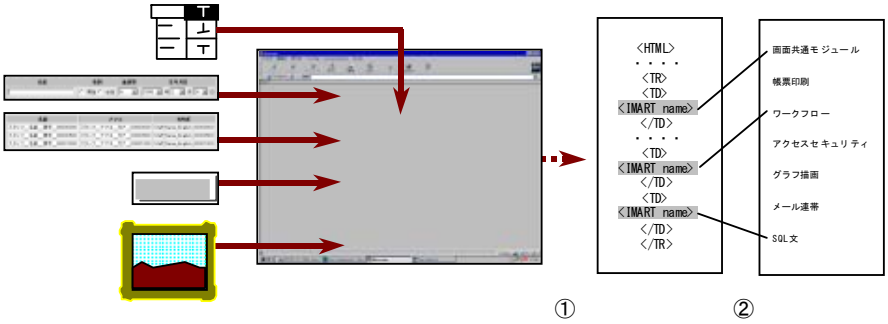
※　im-BizAPI（Java業務コンポーネント）は両モデルから共通で利用することが可能です。

3.2.1 スクリプト開発モデルによるアプリケーション開発

「ファンクションコンテナ」の中には、ビジネスロジックがJavaScriptで記述されており、「プレゼンテーション･ページ」から呼び出され実行されます。その橋渡しの機能をintra-martが実現しています。

3.2.1.1 プレゼンテーションページ

プレゼンテーションページは、ユーザインタフェース部分に相当します。拡張子は「.html」で固定となります。開発者またはエンドユーザは、“eBuilder(Script Producer)”を利用して、Webベースのプレゼンテーションページを作成していきます。
また、ブラウザベースのユーザインタフェースには、動画像やサウンドなどを盛り込んだマルチメディアリッチな画面を構築していくことができます。さらに、プレゼンテーションページはHTMLファイルであるため、Webシステムの開発において、ユーザインタフェース部分のみを切り出してホームページデザイナに作業を依頼することもできます。ホームページ作成ツールから生成されるHTMLファイルに<IMART>タグを追加していくことで、ファンクションコンテナにあるJavaScriptを関連付けて呼び出すことが可能になります。ユーザ定義関数を呼び出す<IMART>拡張タグも追加できます。完成したHTMLファイルは、ページ登録をするだけで、すぐにデータベースと連動して高速動作します。



① 市販のホームページ作成ツール（Microsoft FrontPageやHOMEPAGE BUILDER等）やeBuilder(Script Producer)を利用してユーザインタフェースとなるプレゼンテーションページのひな型を作成します。
② ホームページ作成ツールから自動生成されたHTMLソースに、ファンクションコンテナ中のJavaScript関数やWebPlatformのオブジェクトと連携する<IMART>タグをeBuilder(Script Producer)やテキストエディタを使い挿入（記述）します。

プレゼンテーションページのサンプル例を示します。
intra-martの独自拡張タグ<IMART>を利用して、各種モジュールを呼び出していきます。

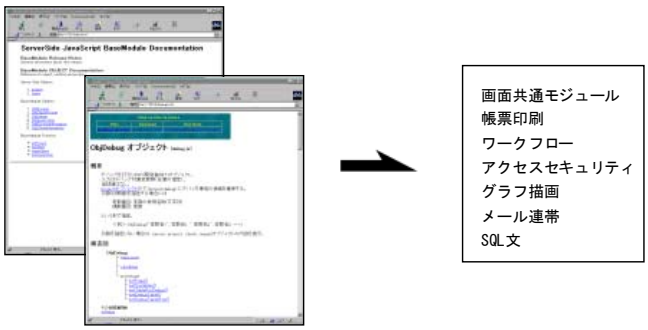
＜社員マスタからのデータ取得用HTML（一覧表示用）ー項目を拡張＞

```
 1: <HTML>
 2: <BODY>
 3: <TABLE border="1">
 4: <TR>
 5:     <TD>社員コード</TD>
 6:     <TD>社員名</TD>
※ 7:     <TD>社員名（カナ）</TD>
 8: </TR>
 9: <IMART type="repeat" list=staffList item="record">
10: <TR>
11:     <TD><IMART type="string" value=record.staff_cd></IMART></TD>
12:     <TD><IMART type="string" value=record.stf_name_kanji></IMART></TD>
※13:    <TD><IMART type="string" value=record.stf_name_kana></IMART></TD>
14: </TR>
15: </IMART>
16: </TABLE>
17: </BODY>
18: </HTML>
```

独自拡張タグ<IMART>を利用して、各種モジュールを呼び出していきます。

3.2.1.2 ファンクションコンテナ

多階層アーキテクチャのうちのApplication Runtime上で稼動するビジネスロジック部分に相当します。拡張子は「.js」で固定となります。ファンクションコンテナとプレゼンテーションページはワンセットとなっているため、ファイルラベル名は同一のものを使用します。開発者は、プレゼンテーションページから呼び出されるJavaScriptを、ファンクション･コンテナの中に記述し作成していきます。具体的には、intra-mart WebPlatformやintra-mart AppFrameworkに用意されているフレームワーク機能から必要なオブジェクトや関数群を選び出し、“eBuilder(Script Producer)”で、それらオブジェクトや関数群を利用したサーバサイドで稼動するビジネスロジックをJavaScriptで記述し作成していきます。データベースへのSQL文もファンクション･コンテナの中に記述していきます。実際のRDBとの接続やSQL発行は、intra-mart WebPlatformやintra-mart AppFrameworkから実行されるため、細かなセッション管理やトランザクション管理を開発者は意識する必要はありません。
作成されたビジネスロジックは、プレゼンテーションページの<IMART>タグから呼び出され実行されることになります。intra-mart WebPlatformやintra-mart AppFrameworkに用意されているフレームワーク機能の詳細は、「intra-mart APIリスト」に一覧記述されています。これらスクリプトの記述はJavaScriptで行えるため、習得が難しいといわれるJavaやActiveXを用いることなく、これまでのホームページ作成の延長でデータベースと連動した本格的なWebシステムの開発が可能になります。

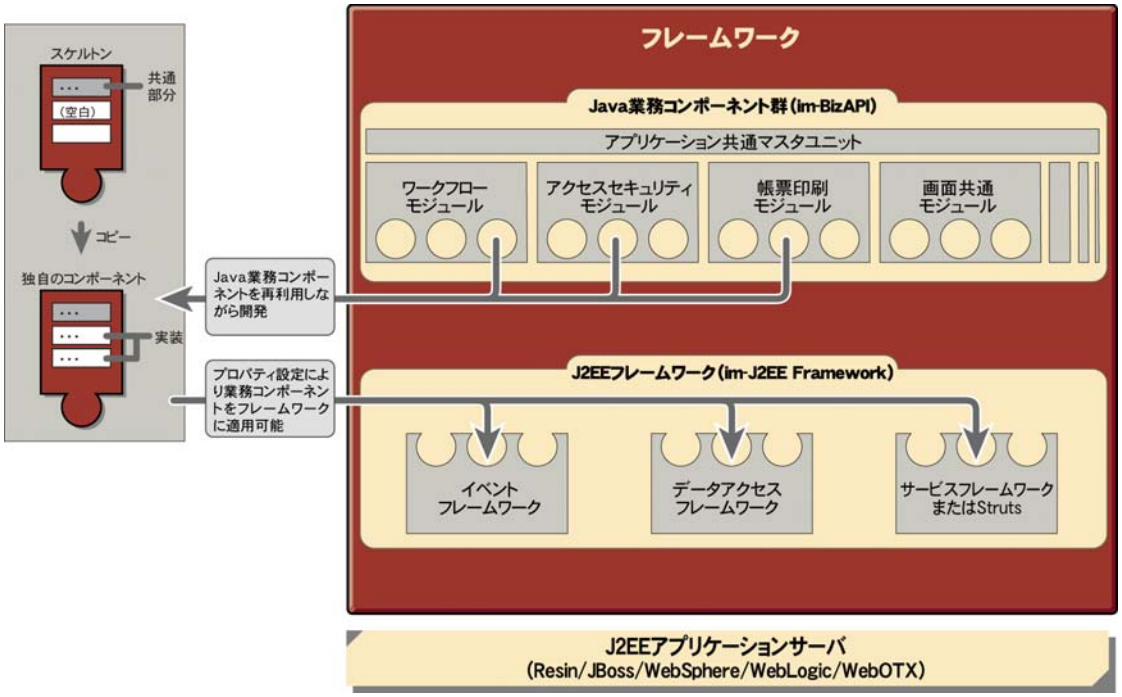


3.2.2 JavaEE開発モデルによるアプリケーション開発

OSやWebアプリケーションサーバに依存しない共有プラットフォームとして、JavaEEによるWebシステム開発が普及してきました。しかし、JavaEEによる開発はJavaをベースにしているため、オブジェクト指向などの高度な知識と経験が要求される点や、JavaEEでの前提知識が必要になる点など、敷居の高さが問題になってきています。さらに、JavaEE開発の規約にさえ準拠させれば、あとはいかようにでも組める自由さが、初心者にとってはかえって負担となり、SEによってバラバラな開発スタイルとなってしまう原因ともなります。
intra-mart WebPlatformやintra-mart AppFrameworkでは、これらの問題をJavaEEフレームワーク（im-JavaEE Framework）を利用することで解決し、JavaEE開発モデルの生産性を大幅に向上させています。

3.2.2.1 JavaEE開発のフレームワーク

JavaEEでのWebシステム開発には、構造的に共通な部分が多く、その事実を利用すると、開発生産性をさらに大きく向上させることができます。intra-mart WebPlatformやintra-mart AppFrameworkでは、JavaEE開発で必要になる共通的な処理は、すべてJavaEEフレームワークとして用意し、開発者に委ねられる箇所はコンポーネントを作成してもらう形態となっています。



<intra-martフレームワークの利用概念>

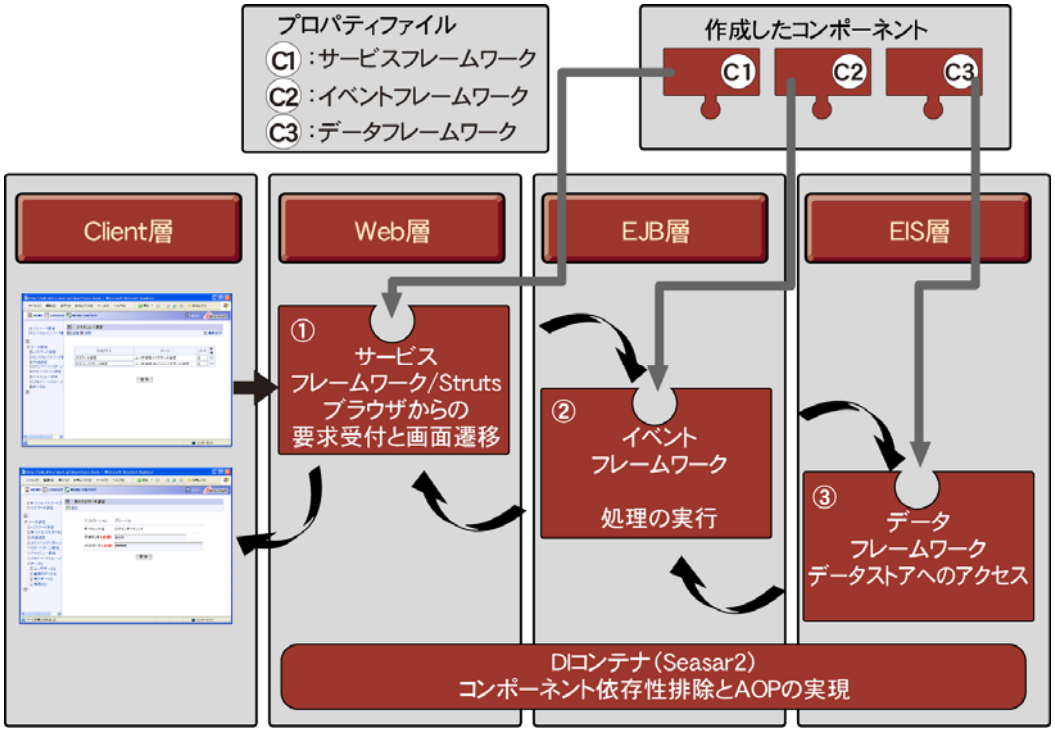
前図のフレームワーク部分にあるように、大きくは「Java業務コンポーネント群（im-BizAPI）」と「JavaEEフレームワーク（im-JavaEE Framework）」に分かれています。Javaコンポーネントモジュール群（ワークフロー・モジュールやアクセスセキュリティ・モジュールなど）はすべて「Java業務コンポーネント群（im-BizAPI）」に配置されます。そしてJavaEE開発で必要になる共通的な処理はすべて「JavaEEフレームワーク（im-JavaEE Framework）」に配置されます。
特に、このフレームワークを利用するコンポーネントについては、雛型となる「スケルトン」が用意されており、さらに生産性を高めることが可能となります（共通部分はあらかじめ実装されているので、開発者は独自のコンポーネントを作成する場合、スケルトンをコピーし、実装されていない箇所を埋めてコンポーネントを作成することになります）。作成されたコンポーネントはプロパティ設定によりフレームワークにあてはめていくことで動作します。

フレームワークもJavaEEで定義されている４層（Client層、Web層、EJB層、EIS層）を使用する共通的な機構となります（SunのBluePrintに準拠）が、開発者はそのことを一切意識する必要はありません（システム規模に合わせてあとから構成変更することも、設定ファイルの変更のみで可能です）。

intra-martのフレームワーク機能は、以下のフレームワークから構成されています。

① サービスフレームワーク	Client層のブラウザからの要求受付の画面遷移（Web層） （下記注を参照してください）
② イベントフレームワーク	主にデータアクセスなどの処理の実行（EJB層）
③ データフレームワーク	データストア（EIS層）へのアクセス
④ DI コンテナ	コンポーネント依存性排除とAOPの実現（Seasar2を組み込み）

これらのフレームワークが、プロパティに設定された内容に従い、該当するコンポーネントを呼び出します。これにより、WebシステムにJavaEEのMVCモデルで処理が実行されることになります。

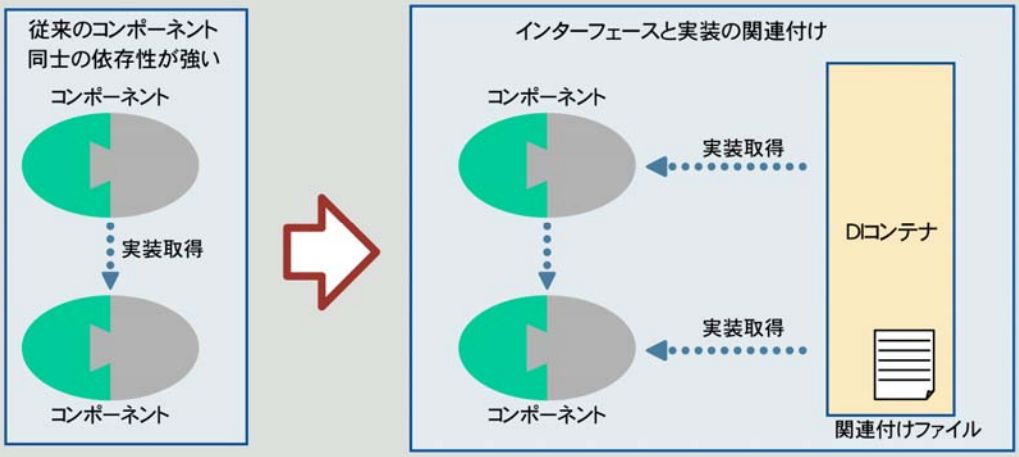


これらのフレームワーク機能を適用することで、開発するシステムを、コンポーネント群とプロパティファイルの設定をベースとした構造に分けることができます。その結果、システムは柔軟にカスタマイズや機能追加できるようになります。

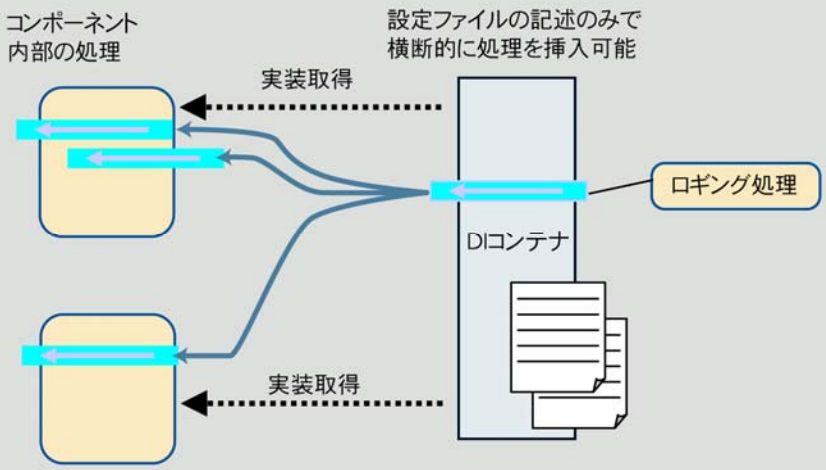
Column

DIコンテナとは

DI コンテナは，コンポーネント間の依存性を内部から排除し、DIコンテナ側で依存関係を構成します。



作成したコンポーネント群に対して「後付け」で機能や処理を挿入できるアスペクト指向プログラミング（AOP）が可能です。



3.2.2.2 JavaEE開発モデルにおいてフレームワークを活用した際のメリット

JavaEE開発時にJavaEEフレームワークを利用することで以下のメリットがでてきます。

JavaEEベースの開発基盤	高度な知識が必要となる部分は隠蔽し、開発者はアプリケーションロジックをスケルトンの中に埋め込む。前提知識がなくても完成したシステムはJavaEEモデルの推奨型となり、MVCモデルの実現が容易(プログラム構造が統一できメンテナンス性向上)。
生産性の向上	共通的なものはすべてフレームワーク中に用意されているので、コンポーネントの再利用性が高まりチーム全体の生産性が向上(コンポーネントのチーム共有と並行分散開発による期間短縮)。
保守性の向上	コンポーネントの新規追加時にも元のアプリケーションにはまったく変更が入らない仕組み。また機能変更時も該当コンポーネントのみの修正とし、他の箇所

への影響がない機構となる。

<フレームワークの適用の有無によるカスタマイズ範囲の違い：○は、影響のある箇所>

順番	変更内容	フレームワーク適用：無			フレームワーク適用：有		
		JSP	Servlet	Bean	JSP	Servlet	Bean
画面カスタマイズ	画面レイアウト変更	○			○		
	画面表示項目追加	○	○	○	○		○
	画面表示形式変更	○	○	○	○		
	画面表示項目削除	○	○	○	○		
	画面フレーム分割	○	○	○	○		
	画面フレーム統合	○	○	○	○		
業務処理アドオン	業務処理追加	○	○	○			○
	業務処理変更	○	○	○			○
DBカスタマイズ	DB項目追加		○	○			○
	DB項目削除		○	○			○

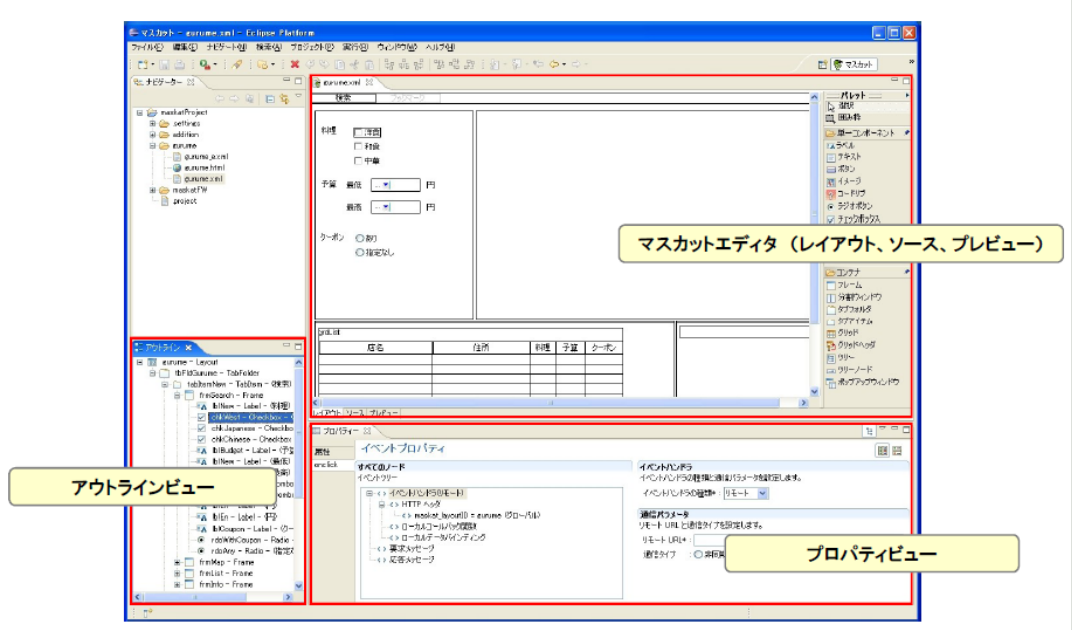
テスト工程期間の短縮と品質向上

テスト工程でフレームワーク部分の確認が不要、また問題発生時の切り分けが容易。またDIコンテナにより、テスト用のモック（模型，仮実装）と正式コンポーネントの入れ替えが簡単になるため，楽にモックを使えるようになる。

Column

Ajaxアプリケーション開発ツール「マスカット」

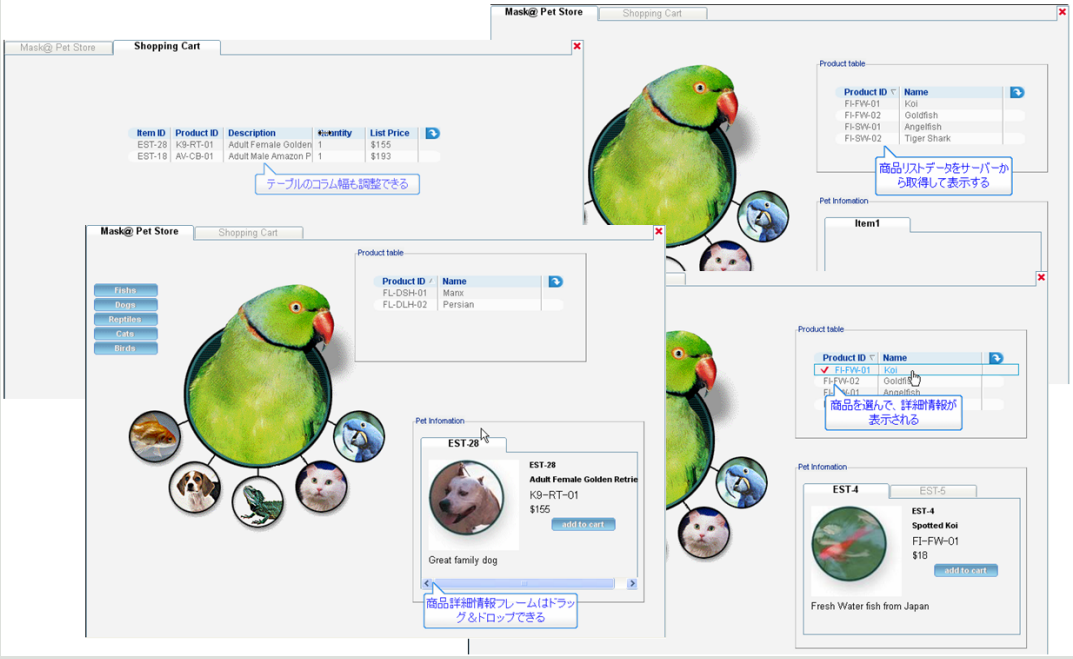
マスカットは、Ajaxベースのリッチクライアントを開発するオープンソースフレームワークで、NTTデータが公開しています。intra-martでは、マスカットが組み込まれています。ユーザは、マスカットを利用することで、intra-mart環境で動作するリッチクライアントを容易に開発することができます。



<マスカット開発画面>

intra-martでマスカットを利用するのメリット

・intra-martでAjaxを利用したリッチクライアント画面を簡単開発可能
・JavaEE開発モデル/スクリプト開発モデルのどちらでもマスカットと連携可能
・画面に配置した各部品の複雑な動作や難解な通信制御を全て隠蔽化
・開発者は定義ファイルの設定とビジネスロジックの作成だけでリッチクライアントを実現



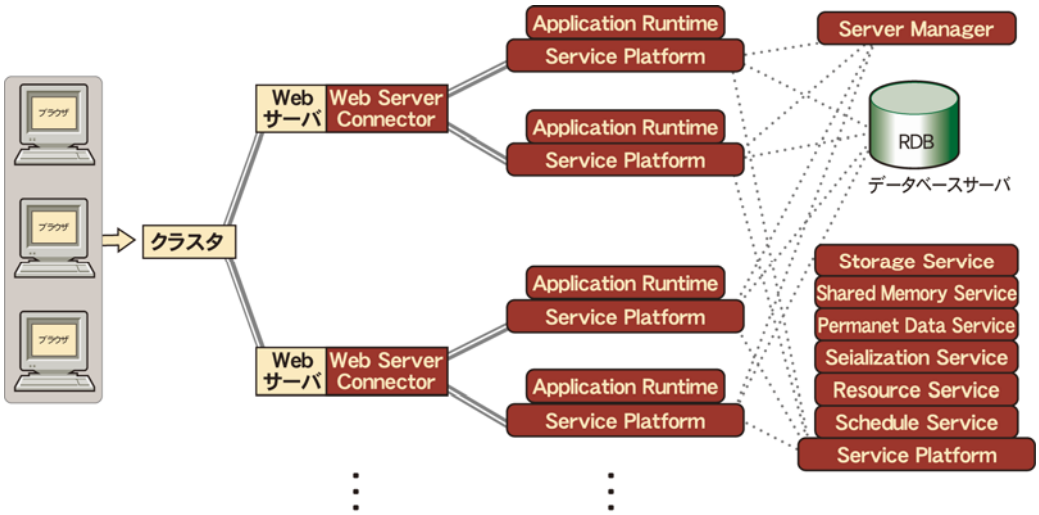
<マスカットを利用して開発した実装例>

3.3 intra-martのシステムアーキテクチャー

intra-mart WebPlatformあるいはintra-mart AppFrameworkのシステムアーキテクチャは以下のようになります。

3.3.1 intra-mart WebPlatformのシステムアーキテクチャー

以下に、intra-mart WebPlatformのシステムアーキテクチャーを示します。



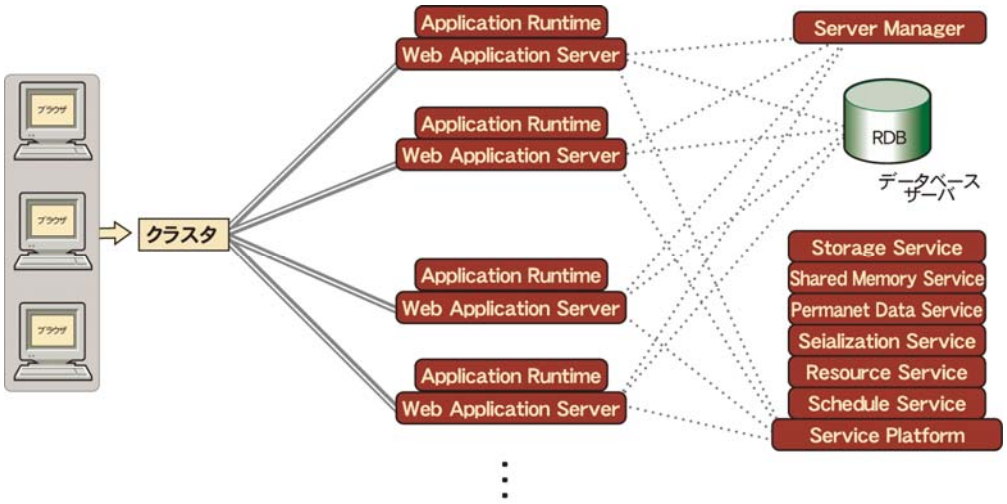
<intra-mart WebPlatformのシステムアーキテクチャー：4台の負荷分散構成例>

Web Server Connector :Webサーバ接続モジュール	Webサーバに組み込むモジュールです。WebブラウザからのユーザのリクエストをwebサーバからService Platformに送ったり、Service Platformの実行結果を受け取りWebサーバに渡すなど、Webサーバとintra-mart Service Platformのインタフェースとして機能します。複数のService Platformがある場合にはラウンドロビン機能により自動振り分け処理もおこないます。
Service Platform :各種サービスのプラットフォーム	intra-martの各種サービスを実行するプラットフォーム。
Application Runtime :アプリケーション実行	Web Server Connectorと連携してWebアプリケーションを実行するランタイム・サービスです。スタンドアロン型で動作させる場合、Application Runtime自身がHTTPサーバとしても機能します(Webサーバは必要ありません)。 Application Runtime機能を利用するには、Schedule Service以外のすべてのサービスが正しく動作していることが必要になります(Schedule Serviceはバッチ管理機能を利用する場合にのみ必要になります)。 スクリプト開発モデルでは、プログラムファイル(プレゼンテーションページ(*.html)およびファンクションコンテナ(*.js))は、Resource Serviceに保管されており、必要時に配信され、Application Runtimeで実行されます。JavaEE開発モデルでは、プログラムファイルはResource Serviceを利用せず、URLによりマッピングされたプログラム(*.jsp、*.xtp、*.class)をロードして実行します。 Application Runtimeは、JDBCを利用して複数のデータベースに接続することもできます。また、構築するアプリケーションの負荷、クライアント数などに応じてApplicatio Runtimeなどを柔軟に増設することができます。複数のApplication Runtimeがある場合は、サーバトラブル発生時においても、並列に配置したApplication Runtimeが相互に補間することができます(セッションフェイルオーバー機能)。また、開発時には当サーバ内にHTTP Serverが内包されているので、別途Web Serverを用意する必要はありません。
Resource Service :プログラム配布	intra-martシステムの動作に必要な各プレゼンテーションページ(*.html)およびファンクションコンテナ(*.js)のプログラムファイルを集中管理すると共に、

	Application Runtimeが必要なプログラムを自動的に配布します。 スクリプト開発モデルのみで利用されるサービスです。
Shared Memory Service :メモリへのデータ保管	データを一時的に保存しておく目的で利用される機能です。また、このサービスが提供する情報管理は、セッションレスです。タイムアウトなどによりデータが消失することはありません。これらの情報は、Shared Memory Serviceのメモリ中に保管され、プログラムの要求に応じて取り出したり削除したりします。
Permanent Data Service :システム設定情報管理	intra-martシステムが動作するために必要な設定情報などを管理しています。 このサービスでは情報をファイル保存するので、サービスが停止した場合も、次の起動時に、以前に保存されたデータを復元することができます。したがって、簡易的なデータベースとして利用することも可能です。また、アカウントやメニューなどの、intra-martのシステム設定情報もこの機能を利用しています。 情報ファイルは、Permanent Data Serviceのインストールディレクトリ直下のtreasure/ディレクトリ内に保存されます。
Serialization Service :システム全体の一意性保証	システム全体における統括的な利用を目的とした機能の集合体です。アプリケーションロック機能の管理が行われています。
Storage Service :ファイルの一元管理	複数のApplication Runtimeを用いた分散システムでintra-martを利用しているときに、アップロードされたファイルやシステムで共有化したいファイル(主にデータファイル)を一元管理するときに利用します。その他、エクステンション・モジュールを用いる場合には、PDFファイルの作成も、このサービスが受け持ちます。ファイルの保存場所は、Storage Serviceをインストールしたディレクトリ直下のstorage/以下になります。
Schedule Service :バッチ実行スケジュール	バッチプログラムの実行スケジュールを管理します。 このサービスは、指定の時間になると該当するバッチプログラムの実行を指定URLを用いてApplication Runtimeに依頼します。したがって、バッチプログラムは、Application Runtimeで実行されます。
Server Manager :動作状況のモニタリング	intra-martの各種サービス、Application Runtimeなどの動作状況を管理します。 標準で用意されている監視ツール「IM-Administrator」を経由して、intra-martの各種Serviceの状況監視や起動・停止を行うことができます。

3.3.2 intra-mart AppFrameworkのシステムアーキテクチャー

以下に、intra-mart AppFrameworkのシステムアーキテクチャーを示します。



Service Platform :各種サービスのプラットフォーム	intra－martの各種サービスを実行するプラットフォーム。
Web Application Server :アプリケーション実行	Web Applicationを実行可能なプラットフォームです。JavaEEの規格に準拠しているサーバ製品をご利用ください（正式に対応している製品に関しては、製品に付属のリリースノートをご覧ください）。
Application Runtime :アプリケーション実行	Application Runtime機能を利用するには、Schedule Service以外のすべてのサービスが正しく動作していることが必要になります（Schedule Serviceはバッチ管理機能を利用する場合にのみ必要になります）。 スクリプト開発モデルでは、プログラムファイル（プレゼンテーションページ（*.html）およびファンクションコンテナ（*.js））は、Resource Serviceに保管されており、必要時に配信され、Application Runtimeで実行されます。 JavaEE開発モデルでは、プログラムファイルはResource Serviceを利用せず、URLによりマッピングされたプログラム（*.jsp、*.class）をロードして実行します。Application Runtimeは、JDBCを利用して複数のデータベースに接続することもできます。 また、構築するアプリケーションの負荷、クライアント数などに応じてApplicatio Runtimeなどを柔軟に増設することができます。
Resource Service :プログラム配布	intra-martシステムの動作に必要な各プレゼンテーションページ（*.html）およびファンクションコンテナ（*.js）のプログラムファイルを集中管理すると共に、Web Application Serverが必要なプログラムを自動的に配布します。 スクリプト開発モデルのみで利用されるサービスです。
Shared Memory Service :メモリへのデータ保管	データを一時的に保存しておく目的で利用される機能です。 また、このサービスが提供する情報管理は、セッションレスです。タイムアウトなどによりデータが消失することはありません。これらの情報は、Shared Memory Serviceのメモリ中に保管され、プログラムの要求に応じて取り出したり削除したりします。
Permanent Data Service :システム設定情報管理	intra-martシステムが動作するために必要な設定情報などを管理しています。 このサービスでは情報をファイル保存するので、サービスが停止した場合も、次の起動時に、以前に保存されたデータを復元することができます。したがって、簡易的なデータベースとして利用することも可能です。また、アカウントやメニューなどの、intra-martのシステム設定情報もこの機能を利用しています。 情報ファイルは、Permanent Data Serviceのインストールディレクトリ直下のtreasure/ディレクトリ内に保存されます。
Serialization Service :システム全体の一意性保証	システム全体における統括的な利用を目的とした機能の集合体です。アプリケーションロック機能の管理が行われています。
Storage Service	複数のWeb Application Serverを用いた分散システムでintra-martを利用してい

:ファイルの一元管理	るときに、アップロードされたファイルやシステムで共有化したいファイル(主にデータファイル)を一元管理するときに利用します。その他、エクステンション・モジュールを用いる場合には、PDFファイルやXRFファイルの作成も、このサービスが受け持ちます。ファイルの保存場所は、Storage Serviceをインストールしたディレクトリ直下のstorage/以下になります。
Schedule Service :バッチ実行スケジュール	バッチプログラムの実行スケジュールを管理します。 このサービスは、指定の時間になると該当するバッチプログラムの実行を指定URLを用いてWeb Application Serverに依頼します。したがって、バッチプログラムは、Web Application Serverで実行されます。
Server Manager :動作状況のモニタリング	intra-martの各種サービス、Application Runtimeなどの動作状況を管理します。 標準で用意されている監視ツール「IM-Administrator」を経由して、intra-martの各種Serviceの状況監視や起動・停止を行うことができます。

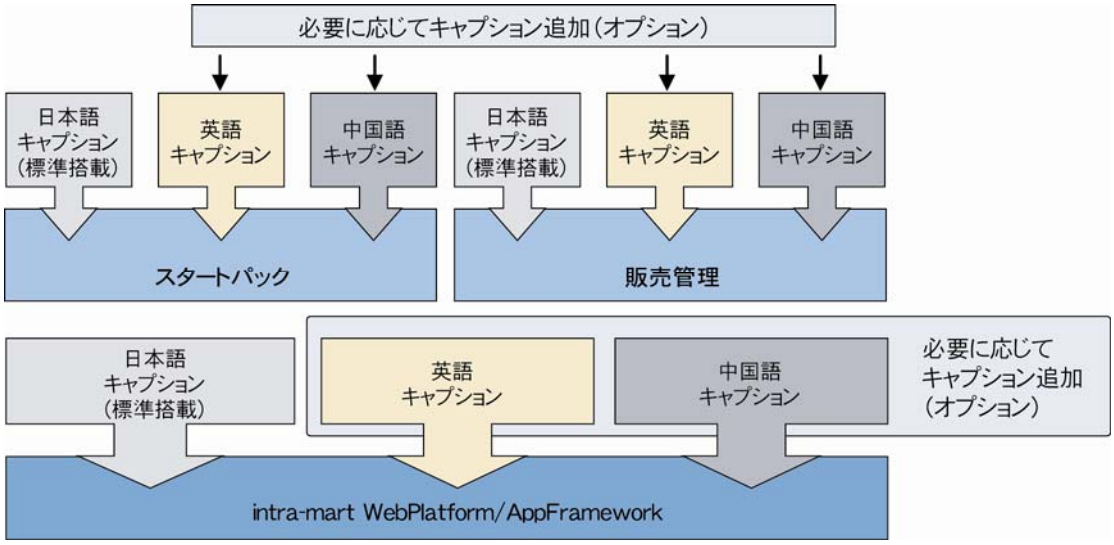
3.4 intra-martのその他の特徴

intra-mart WebPlatform/AppFrameworkには、次のような優位性があります。

- 多言語機能
- 運用の容易性
- 強固なセキュリティ
- ASP 型アプリケーションの構築
- 製品体系と構成エディション

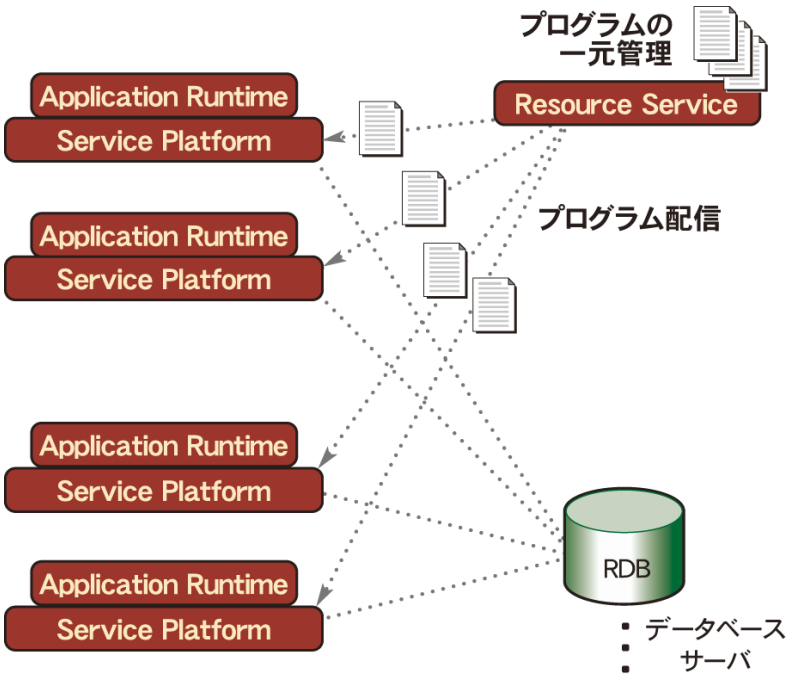
3.4.1 多言語対応

ログインユーザごとに、使用する言語を切替えることができるアプリケーションの開発が行えます。言語を切り替えると、メニュー項目名などがユーザの指定した言語に切り替わります。



3.4.2 運用の容易性

Application Runtime（intra-mart WebPlatformの場合）やWebアプリケーションサーバ（intra-mart AppFrameworkの場合）を複数設置して稼動される大規模システムで問題となるのは、ユーザアプリケーションの配布管理です。スクリプト開発モデルでは、Resource Serviceが各アプリケーションサーバへのユーザアプリケーションの配布を自動で行うため、Resource Service上でのみユーザアプリケーションを最新に保てばよいことになり、運用の負担がかかりません。ただし、JavaEE開発モデル（JSP、Servlet、EJBコンポーネント）の場合は、intra-mart WebPlatformでは各Application Runtime上に、intra-mart AppFrameworkではWebアプリケーションサーバ上に配置する必要があります。
Application RuntimeやWebアプリケーションサーバの新規増設時にも、ネットワークにマシンを接続し、intra-martを起動するだけで最新の稼働環境になります。
また、ブラウザに送信する文字コードをintra-martアプリケーションサーバ上で指定できるため、Windows、UNIXや携帯電話などの機種混在時にも文字化けがおこりません。
さらに、監視ツールであるIM-Administrator（標準装備）により、intra-mart各種サーバの稼動情報をビジュアルに表示でき、各サーバの状態監視や設定変更、自動復旧の設定が遠隔で可能になります。



<システム構成拡張例：intra-mart WebPlatformの場合>

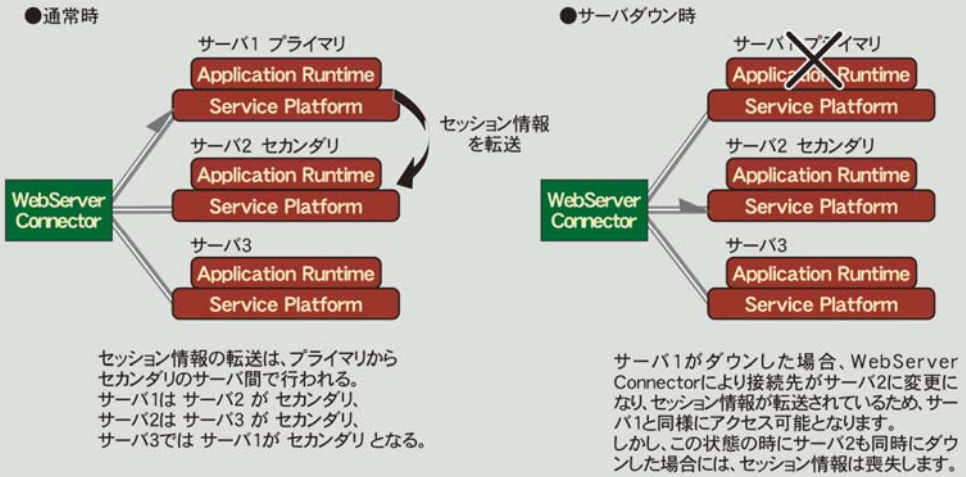
- 詳細については「システム管理者 操作ガイド」を参照してください。

Column

HttpSessionのフェールオーバー

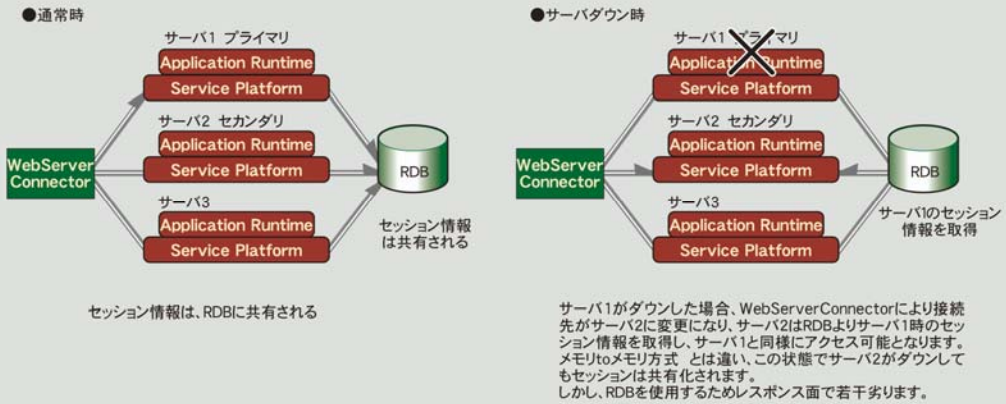
Application Runtimeを複数立てた場合、HttpSession（クライアントのログインセッション情報）をフェールオーバーさせることができます。この機能は、intra-mart WebPlatformで利用できる機能で、intra-mart AppFrameworkでは利用できません（各アプリケーションサーバの機能に依存します）。WebPlatformでのセッションフェイルオーバーは、以下の２方式となります。
詳細については、「intra-mart WebPlatfrom設定ガイド」の「HttpSessionのフォールトトレランス設定方法」を参照してください。

● メモリtoメモリ方式



メリット　Application Runtimeの設定だけで構築可能。メモリto RDBと比較して、処理負担がない。デメリット　　バックアップ関係にある２台が両方ともダウンするとセッションが消える。メモリ to RDB方式に比べて設定が難。他のApplication　Runtimeのメモリを保存するため、Application　Runtimeのメモリ消費量が増える。

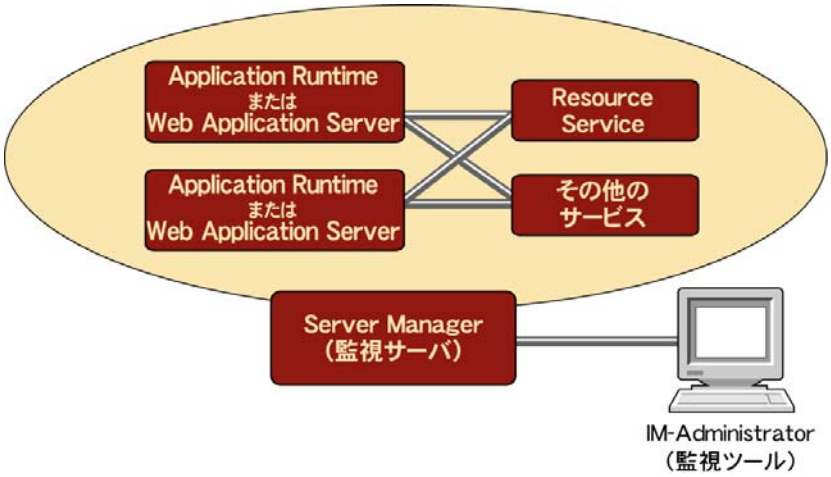
● メモリto RDB方式



メリット　Application Runtimeが何台ダウンしても最低１台残ればセッションの継続が可能。
メモリ to メモリ方式に比べて設定が容易。
デメリットＤＢが必要。
市販のクラスタ製品を利用してラウンドロビン構成で運用する場合には、メモリ to RDBを推奨します。より堅牢なシステム構成にするには、メモリ to RDB方式を推奨します。

3.4.2.1 IM-Administrator

監視サーバ（Server Managerに含まれる機能）によって収集されたintra-martの各種サービスの稼動情報をビジュアルに表示する監視ツール（標準装備）です。各サーバの状態の監視や設定の変更、自動復旧の設定が遠隔で行えます。



<IM-Administrator（監視ツール）の基本画面>

3.4.3 強固なセキュリティ

■アクセスセキュリティモジュールでの認証

情報漏洩対策として、システム利用者を制限します。

- ログインユーザの認証機能（LDAP連携可能）
- アクセスコントロール 《個人レベルの利用制限》
- ログイングループ設定《組織レベルの利用制限》

■アクセスログの保存

情報流出時の追跡･監査として、誰が、何時、システムを使いどのような操作をしたかを記録します。
保存方法として、ファイルおよびデータベースに保存することができます。
データベースに保存されたログは、View Creatorを使用して参照することも可能です。

- アクセスログにてアクセス元を特定することが可能
- ログイン／ログアウト履歴の保存
- ユーザの操作ログ保存

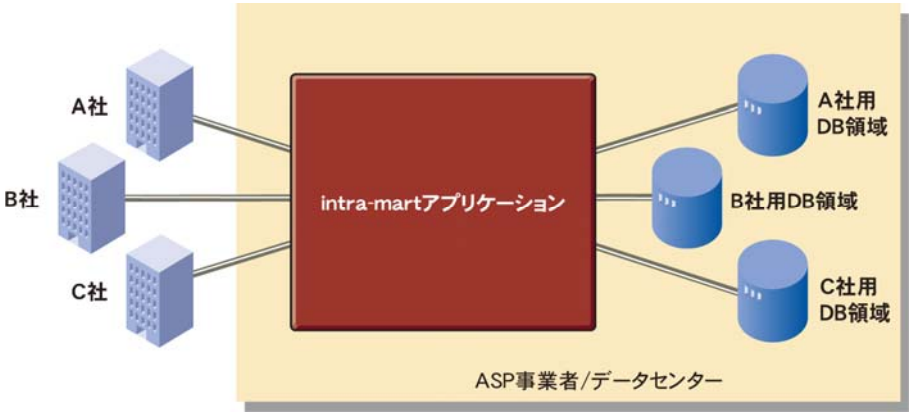
■IM-SecureBlocker（エクステンションシリーズ）の利用

Webアプリケーションの脆弱性対策として、外部からの不正な攻撃を遮断します。

- SQL インジェクション、クロスサイドスクリプティングによる攻撃を防御

3.4.4 ASP型アプリケーションの構築

ログインするユーザグループごとに、ロール、ユーザ、メニュー、リンク、メッセージそしてデータベースを異なる設定で利用することができます。この機能を利用すると、ひとつのアプリケーションを複数の会社で共同利用するようなASP型（マルチシェア型）のアプリケーションを設定だけで簡単に構築することができます。



intra-martのすべての設定は、グループや会社ごとに個別に設定することができ、それぞれのグループが個別のintra-martシステムを利用しているように使用することができます。また、マルチデータベース機能と合わせて使用すると、それぞれのグループで別のデータベースサーバを利用することができます。マルチグループ機能を設定した場合は、ログイン画面にマルチグループIDを入力する項目が表示されるようになります。

3.5 intra-mart「eBuilder」

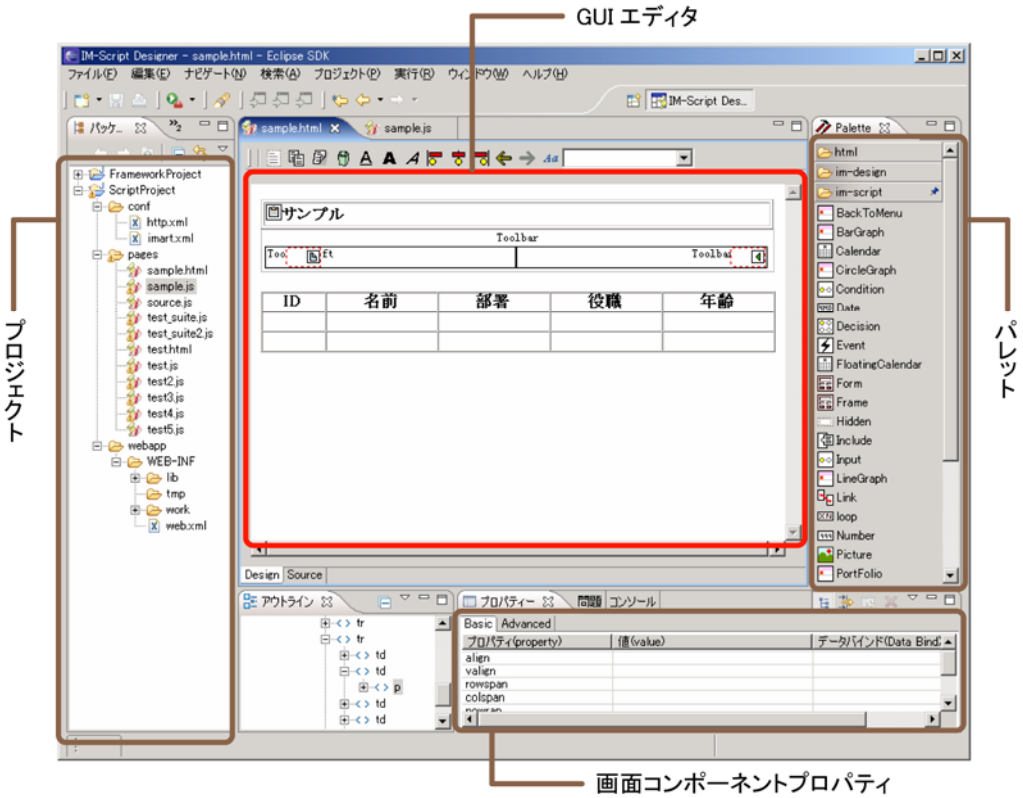
eBuilderはオープンスースの統合開発環境である「eclipse」に対するプラグインとして実装されています。eclipse が持つ機能とeBuilderとしての機能の両方を利用することで、開発効率が大きく向上します。また、「スクリプト開発モデル」と「JavaEE開発モデル」の両方を同時にサポートしていますので、あらゆる開発シーンにおいて統合開発環境としてご利用いただけます。

- intra-mart WebPlatform/AppFrameworkには、intra-mart「eBuilder」の試用版が同梱されています。
  ぜひ、一度体験することをお勧めいたします。

3.5.1 Script Producer

スクリプト開発モデル用のScript Producerには、さまざまな開発支援機能が搭載されており、HTML、JavaScriptを利用したプレゼンテーションページとファンクションコンテナによる開発効率を大きく向上させるツールです。

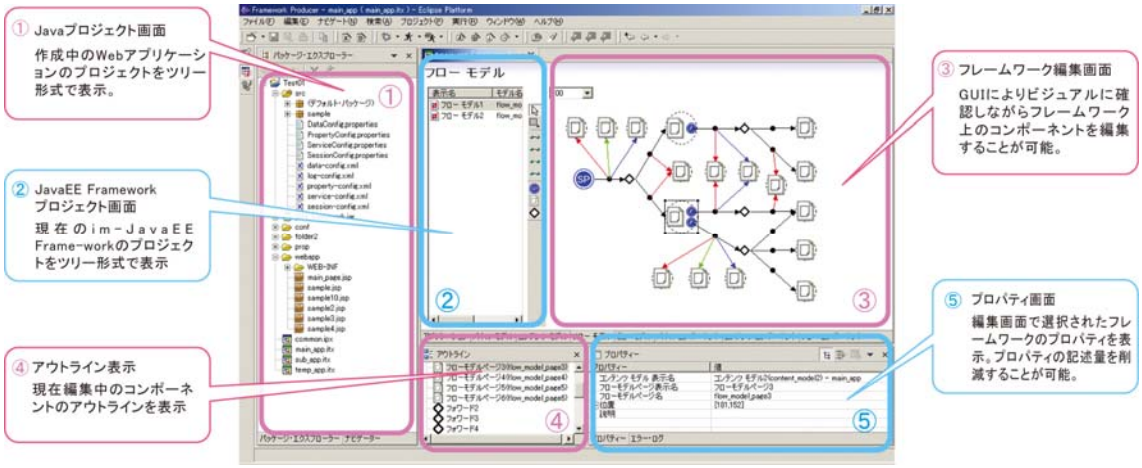
- 画面作成に GUI エディタを提供 — 各種画面コンポーネントを用意
- JavaScript でのコーディングを強力にサポート — API 入力支援、リアルタイム構文チェック
- 実行時デバッグ環境の提供 — トレース実行、変数の参照・・・
- 単体テスト環境の提供（im-JsUnit） — テストケースもすべて JavaScript のみで完結
- マスカット IDE との連携機能を提供 — Client および Server 側のロジックを統合的に開発



ボタンや表、カレンダー、グラフ、HTML 標準のタグなど、パレットに用意されている画面コンポーネントを GUI エディタ上にドラッグ＆ドロップし、実際のレイアウトを確認しながら画面を作成できます。ソース編集による開発も可能です。

3.5.2 FrameworkProducer

JavaEEベースでの開発に最適な開発ツールです。intra-mart　JavaEE Frameworkで使用される設定ファイル（プロパティファイル群）をGUIツールによって自動的に作成したり、スケルトンコードを自動生成することができます。JavaEEフレームワークの編集がビジュアルに可能となるオーサリングツールが提供されることで、煩雑なJavaEE開発の効率を大きく向上させることができます。

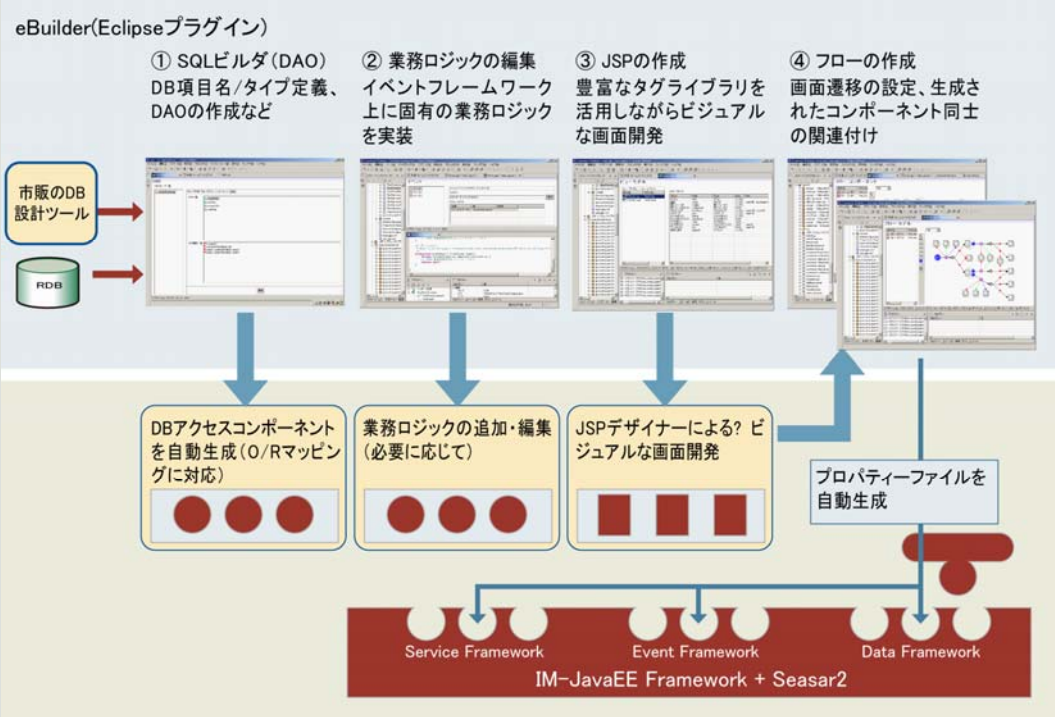


3.5.3 FrameworkProducerのソースコード自動生成、ビジュアルな画面定義機能

FrameworkProducerには、画面関連のコンポーネント、業務ロジック、DB関連のコンポーネントなどを自動生成する機能が標準で提供されています。各編集画面から必要な項目などを入力するだけで、JavaEEのソースコードを生成することが可能です。生成されたソースコードはそのまま intra-martのim-JavaEE Framework上で稼動するコンポーネントとして利用することができるようになるため、Javaのコーディング経験の浅い開発者でも、Webシステム開発を容易に、効率的に進めていくことができます。

Column

ソースコードの自動生成～実行までの利用手順



① SQLビルダによるDAOの作成
データフレームワーク上で動作するコンポーネントを作成します。こちらではデータベース（DB）構造を定義するもので、市販のDB設計ツールからのデータ取り込みも可能です。
用意された編集画面上で、DBの項目名やタイプなどのテーブル情報を登録し、データベース・クエリをビジュアルに定義することで、DBへのアクセスを行うクラスを自動生成できます（ORマッピングに対応）。

② 業務ロジックの作成
イベントフレームワーク上で動作するコンポーネントを作成します。システム内で利用される業務ロジックをeBuilder上で作成し、部品化していきます。

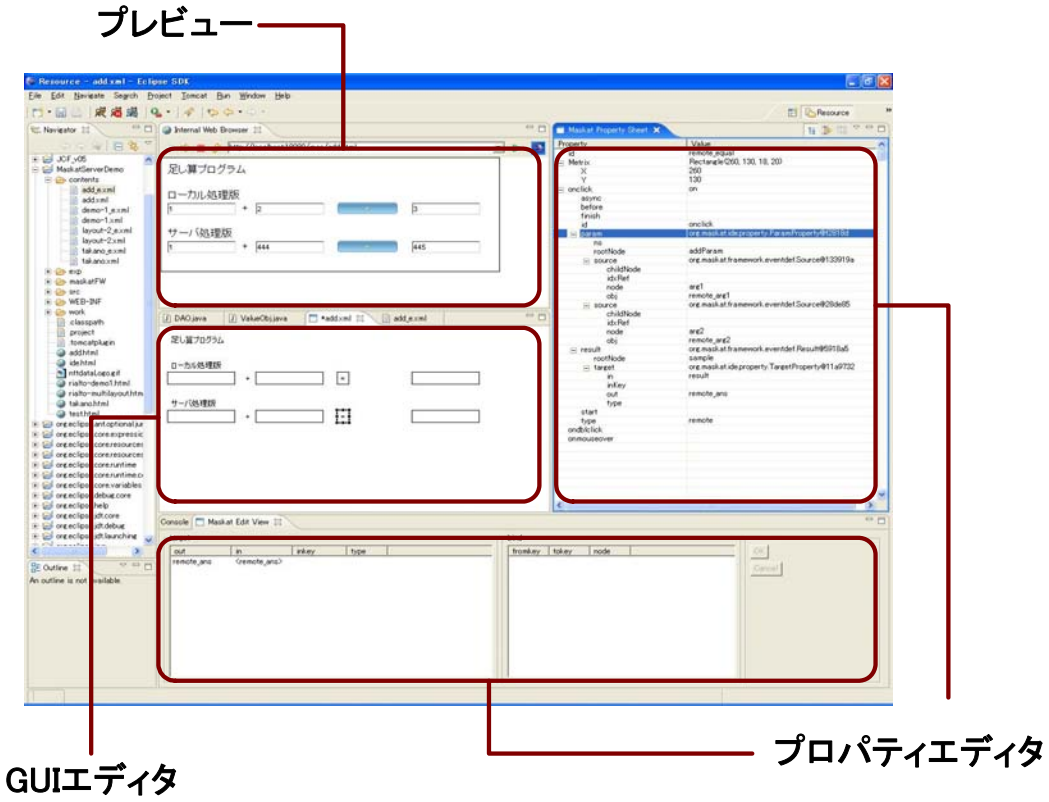
③ JSPの作成
豊富なタグライブラリを活用しながら、ビジュアルに画面を開発することができます。

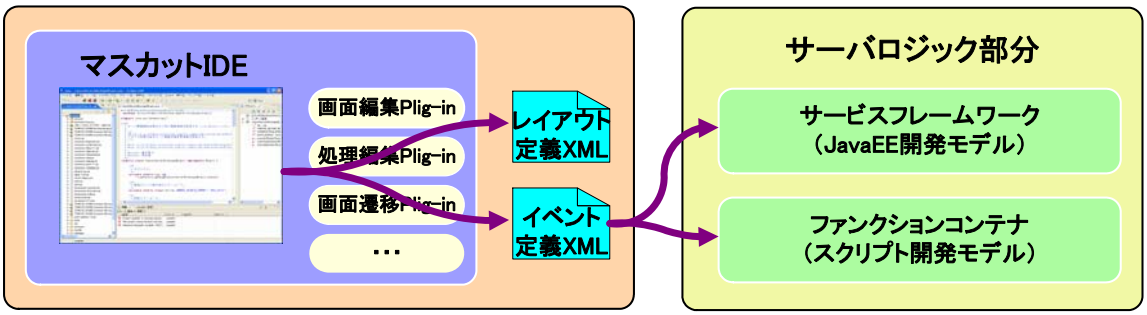
④ フロー作成（画面、画面遷移の作成とコンポーネントとの関連付け）
上記で作成された各コンポーネントを編集画面からビジュアルに呼び出し、コンポーネント同士の呼び出しを定義していきます。ここからプロパティファイルを自動出力します。

⑤ ソースコード完成したコンポーネントをJavaEEフレームワーク上に配置・実行編集・追加
eBuilderのソースコード自動生成機能を利用することで、開発すべき全体ソースコードの70～80%が自動生成されるため、従来よりも開発効率が大幅に向上いたします。

3.5.4 マスカットIDEによるビジュアルな画面定義機能

FrameworkProducerおよびScript Producerには、マスカットIDEが標準で組み込まれています。マスカットでは、開発者は画面を定義する「レイアウト定義XML」と動作を定義する「イベント定義XML」の2種類のXMLファイルを作成するだけでリッチクライアントを構築することができます。intra-martでは「イベント定義XML」と2つの開発モデル（「スクリプト開発モデル」と「JavaEE開発モデル」）と連携が可能です。また、「イベント定義XML」からサーバロジックの雛形が作成されますので、雛形に対して業務ロジックを記述することでリッチクライアントを容易に開発することができます。





3.6 旧バージョンとの互換性

本バージョンには、Ver6.1で作成したアプリケーションとの互換性を保つための仕組みが用意されています。

3.6.1 APIの互換性

APIの互換性に関しては、一部のAPIを除き両Ver間での互換性が保たれています。

- 詳細については、「リリースノート」および「APIリスト」を参照してください。

3.6.2 データベース構造の互換性

データベース構造は一部変更があるため、「データ移行ツール」が提供されています。

3.6.3 旧リビジョンとの互換性（Ver.6.1→Ver.7.0）

Ver.6.1とVer.7.0に関する互換性は、APIは互換性がありますが、データベース構造は変更されています。移行方法に関しましては「移行ガイド」を参照してください。

3.6.4 JumpUpモジュール（Ver.3、4→Ver.5.1~Ver.7.0）

intra-martのVer3およびVer4からVer5.1～Ver7.0にアップグレードするために、JumpUpモジュールが用意されています。JumpUpモジュールでは、次の図中の③と④のアップグレードが可能となります。以下に、Ver3およびVer4からVer5.1～Ver7.0にアップグレードする際の注意点を示します。

① DB移行ではAP共通マスタの移行が可能。

② intra-martベースモジュール（WebPlatform）をカスタマイズしている場合は、新しいVer5.1を再インストールした上で再度カスタマイズする。

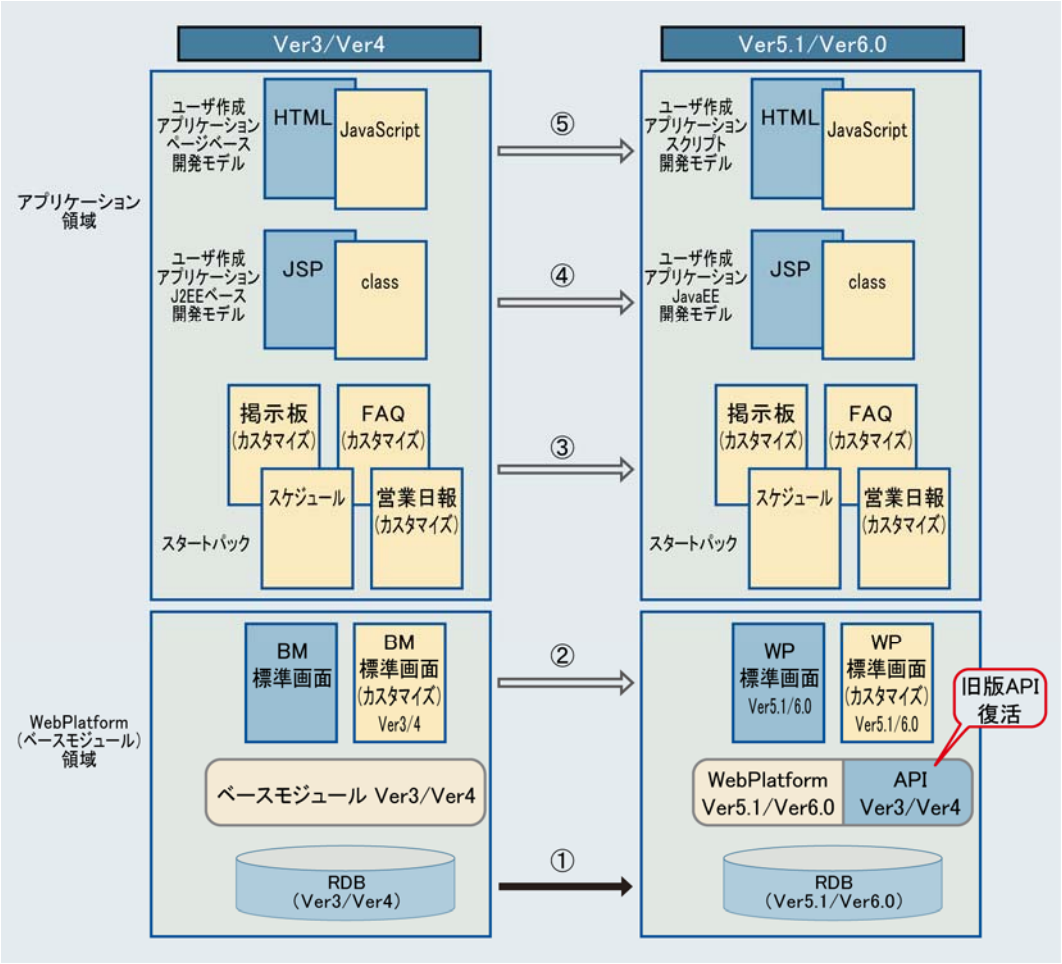
③ Ver3.2以上で作成したユーザ作成のアプリケーションは、Ver5.1～Ver7.0でそのまま移植して動作する（スクリプト開発モデルのみ）。

④ Ver.4.0以上で作成したユーザ作成のアプリケーションは、Ver5.1～Ver7.0にそのまま移植して動作する（im-JavaEE Framework部分のみ）。

⑤ intra-martアプリケーションに関しても、カスタマイズしたアプリケーションをそのままVer5.1～Ver7.0に移植して動作する。

≪注意点≫

- ③④⑤は intra-mart ジャンプ・アップ・モジュールを利用。(別途コンサルティング契約必要)。
- そのままの移植では旧 AP 共通マスタ(履歴対応なし)への対応となります。
- 新 AP 共通マスタ(履歴対応)に対応するためには API の切り替えおよび使用 SQL の変更が必要となります。
- 多言語対応などにともない一部の API は切り替えが必要なものもあります。
- 非推奨 API は対応していないため、非推奨 API を使用していた場合、API の切り替えが必要です。
- DB 移行ではワークフローのトランザクションデータ移行はできませんが、旧ワークフローをそのまま DB 移行することにより継続して旧ワークフローが動作します。
- DB サーバによっては問題が発生する場合があります。
- 環境面については再度サイジングを行う必要があるため、コンサルティングの利用を推奨いたします。
- 一部のAPIはインタフェース互換されておりません。互換できないAPIを使用している箇所は、修正が必要になります。



- JumpUpモジュールに関しては、営業窓口までご連絡ください。

B

BPM **11**

C

CRM ソリューション 3

D

DI コンテナ **67**

E

eBuilder 79

F

FormatCreator **42**

I

IM-Administrator 77
im-BizAPI 24
IM-ERP リアルコネクト **61**
IM-EX 申請システム **58**
IM-PDF デザイナー **54**
IM-SecureBlocker **57**
IM-SecureSignOn 56
IM-SonicESB 59
IM-X Server 55
intra-mart AppFramework 2
intra-mart WebPlatform 2
intra-mart アプリケーションシリーズ 2
intra-mart の特長 4

J

J2EE ベース開発モデル **6**
Java 業務コンポーネント群 25
Java コンポーネント 24
JSR168 ..... **14**
JumpUp モジュール 84

S

Seasar2 HotDeploy **22**

T

TableMaintenance **45**

V

ViewCreator **44**

W

Web サービス **17**

あ

アクセスセキュリティモジュール **33**
アプリケーション共通モジュール **31**
アプリケーション共通マスタ 47

い

イベントナビゲータ **46**
イントラネットソリューション 3

え

ASP 型アプリケーション **78**
エクステンションシリーズ 54

か

外部ソフトウェア接続モジュール **41**
画面共通モジュール **27**
画面デザイン **50**
カレンダ表示モジュール **29**

き

業務基盤ツール 42
業務基盤モジュール 31

く

グラフ描画モジュール **29**

し

システムアーキテクチャー 70

す

スクリプト開発モデル **7**

せ

全体構成 2

つ

ツリー表示モジュール **30**

は

バッチ管理モジュール **40**
パブリックグループ 48

ふ

ファンクションコンテナ 64
プライベートグループ 48
プレゼンテーションページ 63

ほ

ポータルモジュール **12**

ま

マイメニュー 36
マスカット **22**, **30**, **68**

め

メール連携モジュール **40**

ゆ

ユーザインタフェース層 27

ろ

ログイングループ管理者権限の委譲 19
ログ機能 **21**

わ

ワークフローモジュール **38**
ワークフローモジュール **15**

intra-mart WebPlatform/AppFramework Ver.7.0

2010/11/30 第５版

イントロダクション

株式会社 NTT データ イントラマート
〒107-0052 東京都港区赤坂 2-17-22 赤坂ツインタワー本館 3 階
TEL(03)5549-2821　FAX(03)5549-2816
E-mail ： info@intra-mart.jp
ホームページ ： http://www.intra-mart.jp
Copyright 2000-2010 株式会社 NTT データ イントラマート All rights Reserved.

※本マニュアルに記載されている社名および商品名は、一般に各社の商標および登録商標です。