

Fiorano®

Fiorano SOA プラットフォームの アーキテクチャ概要

対象バージョン : 2007 SP7 および 9.0.0

はじめに

この文書は、Fiorano SOA プラットフォームの製品概要、アーキテクチャ、ビジネス プロセスについて基本的な事項を説明するものです。

このガイドブックの他に、以下のガイドブックも用意されています。併せてご参照ください。

- Fiorano SOA プラットフォームのダウンロード (製品インストーラと評価用ライセンスの取得方法)
- Fiorano SOA プラットフォームのインストール
- Fiorano SOA プラットフォームの起動方法
- Fiorano オーケストレータによるビジネス プロセス構築の概要

目次

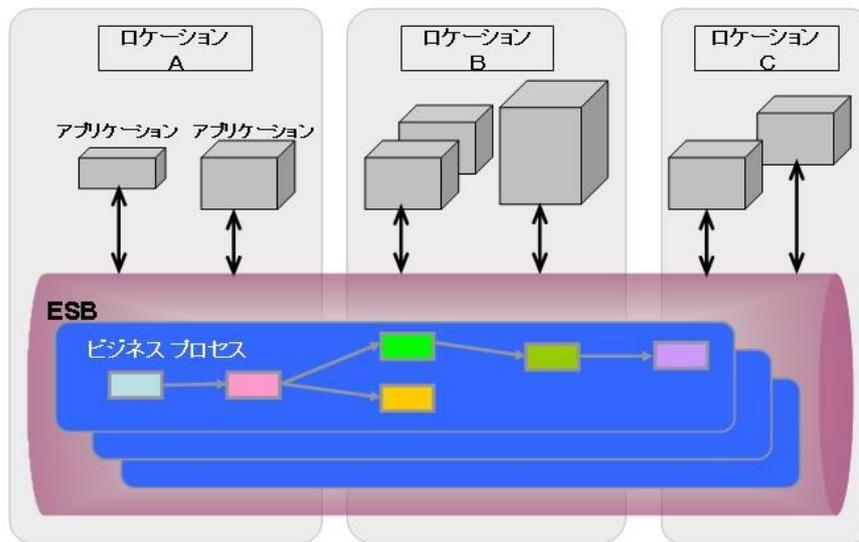
1. Fiorano SOA プラットフォームのアーキテクチャ概要	3
1.1 Fiorano SOA プラットフォームの製品コンセプト	3
1.2 Fiorano SOA プラットフォームのサーバー群	4
1.2.1 ピア サーバー	4
1.2.2 ESB サーバー	5
1.2.3 WebGateway	6
1.3 Fiorano SOA プラットフォームの全体アーキテクチャ	7
2 ビジネス プロセスの概要	8
2.1 ビジネス プロセスのレイヤー構造	8
2.2 サービス コンポーネント	10
2.3 ビジネス プロセス ビューと物理的なデプロイメント	11
3 プリビルト コンポーネントの一覧	14

1. Fiorano SOA プラットフォームのアーキテクチャ概要

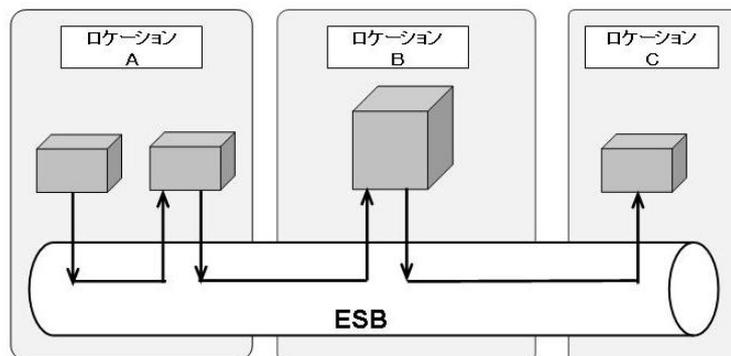
1.1 Fiorano SOA プラットフォームの製品コンセプト

Fiorano SOA プラットフォームは、EAI (アプリケーション インテグレーション)、BPM (ビジネス プロセス マネージメント)、ビジネス プロセスの自動化、B2B インテグレーションなどに対する共通したプラットフォームとして機能します。このプラットフォームの基盤として採用されているのが ESB (エンタープライズ サービス バス) です。Fiorano の ESB は、企業内外に分散する様々なソフトウェア アプリケーションを接続するための広範なコネクティビティと分散環境に対する優れたサポート機能を有しています。また、分散されたアプリケーションの連携をビジネス プロセスとして構築 (オーケストレーション) し、実行するための BPM 機能も備えています。

すなわち Fiorano SOA プラットフォーム のコンセプトとは、取引先企業を含んだ企業内外のアプリケーションを 1 本の ESB によって結合し、アプリケーションを 1 対 1 の関係ではなく複数のステップからなるビジネス プロセスとして構築、実行するプラットフォームを提供することです。



Fiorano SOA プラットフォームでは、各アプリケーションは、中央のサーバーやプロセス エンジン (代表的なものに BPEL エンジンがあります) から呼び出されるような中央制御の方式を採っていません。アプリケーションからアプリケーションに、中央を経由することなく、ビジネス データやビジネス イベントの通知をダイレクトに送信します。これは、中央のハブ (サーバーやプロセス エンジンなど) に負荷も障害も一極集中してしまうことを避ける有効な方法です。ESB (エンタープライズ サービス バス) が“バス”という名称を持っているのは、中央集中を避けるためのデータ通信経路としてのバス構造を本来意味しているからです。



Fiorano SOA プラットフォームでは、ESB が各ロケーションをまたがった 1 本のバスとして機能します。データは、アプリケーションからアプリケーションへ ESB のメッセージング機能を利用して送られます。中央でビジネス プロセスを制御するものではなく、ESB 上でプロセス制御が行われます。これは次に挙げる 2 つのメカニズムによって実現されています。

- ピア サーバーによるピア ツー ピア メッセージング
- コンポーネントによるビジネス プロセス

次のセクションでは、ピア サーバーを中心に、Fiorano SOA プラットフォームのサーバーについて説明します。コンポーネントによるビジネス プロセスについては、次章で説明します。

1.2 Fiorano SOA プラットフォームのサーバー群

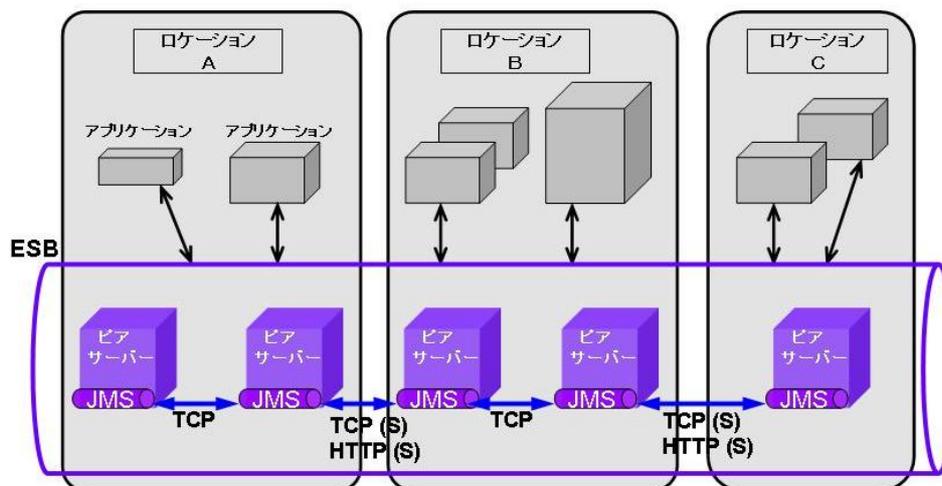
上述のコンセプトを実現するために、Fiorano SOA では 3 種類のサーバーを設けています。

1.2.1 ピア サーバー

下図に示すように、ピア サーバーは、ESB の伝送路を形成するものです。

ピア サーバーの peer という言葉の意味は、各サーバーの機能が同等で、サーバー間に上下関係などの区別がないことを指しています。Peer-to-Peer のコミュニケーションとは、同等のサーバー間でのコミュニケーションを意味し、ハブやブローカーといった何らかの制御を司る中心的なサーバーを介さないことを意味しています。ポイント ツー ポイント (Point-to-Point) と混同されがちですが、これとは異なる意味をもったものです。Fiorano ESB では、分散された拠点にピア サーバーを配置することで、すべての拠点をカバーするメッセージング バスを形成します。このメッセージング バスの上でビジネス プロセスが実行されます。ビジネス プロセスとメッセージング バスとの関連についての詳細は、次章『2 ビジネス プロセスの概要』で説明します。

拠点に配置されたピア サーバー間の転送プロトコルには SSL がサポートされており、ファイヤーウォールによって保護された拠点や取引先企業との間の転送も可能とします。また、各拠点の負荷状況によってピア サーバーを複数配置することができ、柔軟なスケーラビリティを実現しています。



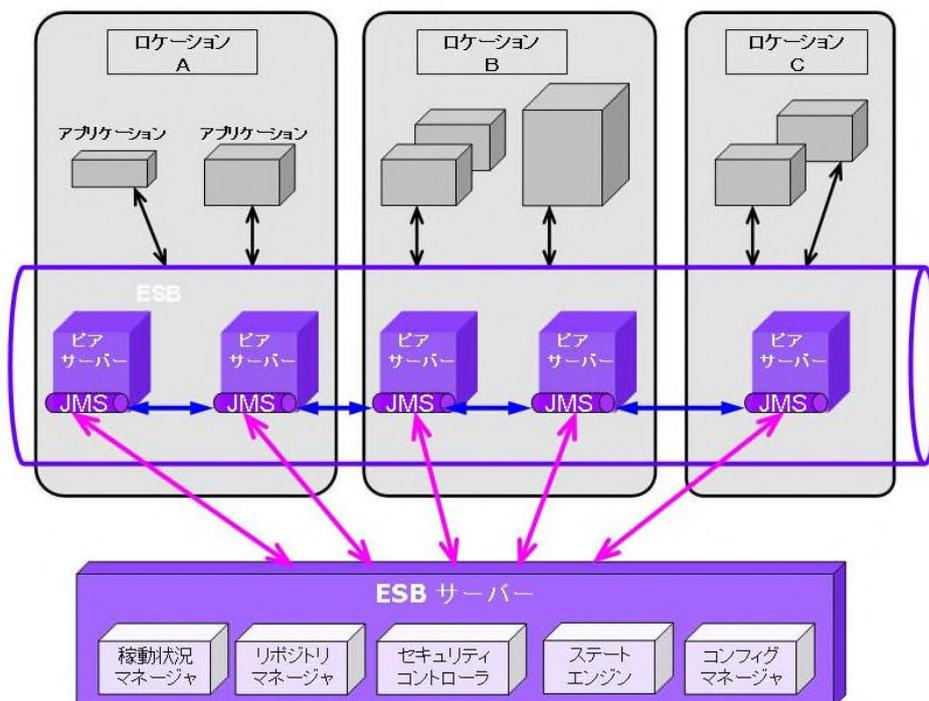
1.2.2 ESB サーバー

ESB サーバーは、ESB 全体 (ピア サーバー群で構成される伝送路) の管理センターとして機能します。

管理機能には、

- サーバーのコンフィグレーション
- セキュリティ設定
- ピア サーバーおよびビジネス プロセスの管理および状態監視とロギング
- ビジネス プロセスのデプロイメント、実行、停止
- サーバー コンフィグレーション、セキュリティ設定情報、ビジネス プロセスなどのメタ情報を保持するリポジトリの維持、管理

などがあります。管理者は、Fiorano Studio ツールやその他の管理ツールを用いてこれらの管理機能を利用します。

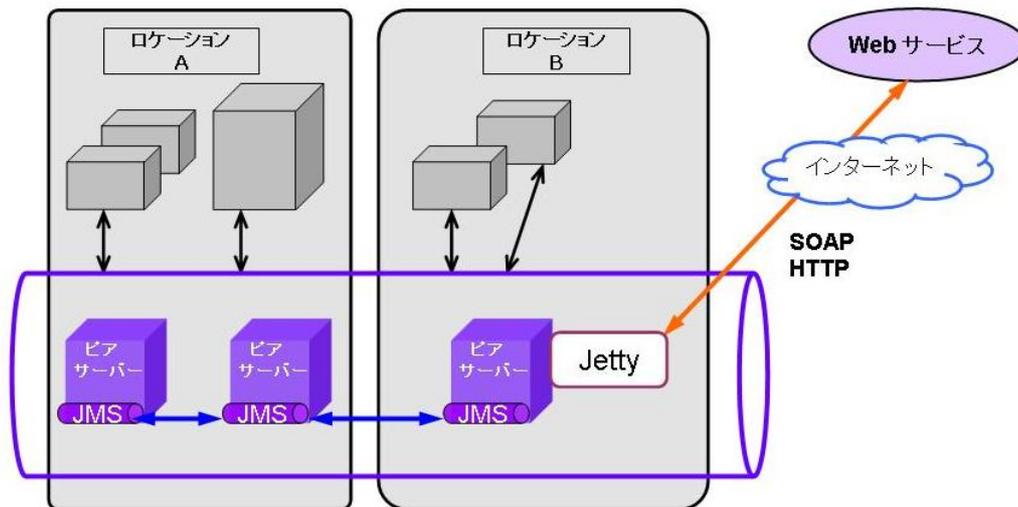


また、ESB サーバーは、ビジネス プロセス構築 (オーケストレーション) の開発環境としての機能も提供します。ビジネス プロセスの構築は、Fiorano Studio ツールを用いて行いますが、このツールは ESB サーバーに接続してリポジトリや ESB サーバーの様々な機能にアクセスします。ビジネス プロセスの開発も、管理機能と同様に中央で一元的に実施されます。

Fiorano Studio のオーケストレーション機能によって構築されたビジネス プロセスは、ESB サーバーのリポジトリに格納されます。リポジトリに格納されたビジネス プロセスは、ESB サーバーのデプロイメント機能によって各ピア サーバーにデプロイされ、実行されます。いったん実行されたビジネス プロセスは、ピア サーバーによって処理され、メッセージ (データ) は、ESB サーバーを経由せずにピア サーバー間の経路上を転送されます。すなわち、上図のピンクの矢印上は、管理コントロールやログ情報などの管理データのみが転送され、ビジネス プロセス本来のビジネス データは、青色の矢印上を流れます。こうすることで、ビジネス データの一極集中を避け、耐障害性の向上や負荷分散の向上を実現しています。

1.2.3 WebGateway

WebGateway (Web ゲートウェイ) は、SOAP メッセージのエンコーディングと HTTP (SOAP over HTTP) プロトコルを介したデータの送受信を行います。Fiorano SOA 2007 SP4 以降のバージョンでは、Jetty がピア サーバーに組み込まれるようになりました。この Jetty が WebGateway をホストする J2EE コンテナとして使用されます。旧バージョンでは、Tomcat が WebGateway をホストするデフォルトのアプリケーション サーバーとして製品にバンドルされており、独立したサーバーとして稼働させていました。2007 SP4 以降のバージョンでは、Tomcat に代わって Jetty がピア サーバー内に組み込まれており、ピア サーバーが起動されると WebGateway (Jetty) も連動して起動されます。ユーザーは、WebGateway や Jetty について特別な操作やコンフィグ設定を行う必要はありません。



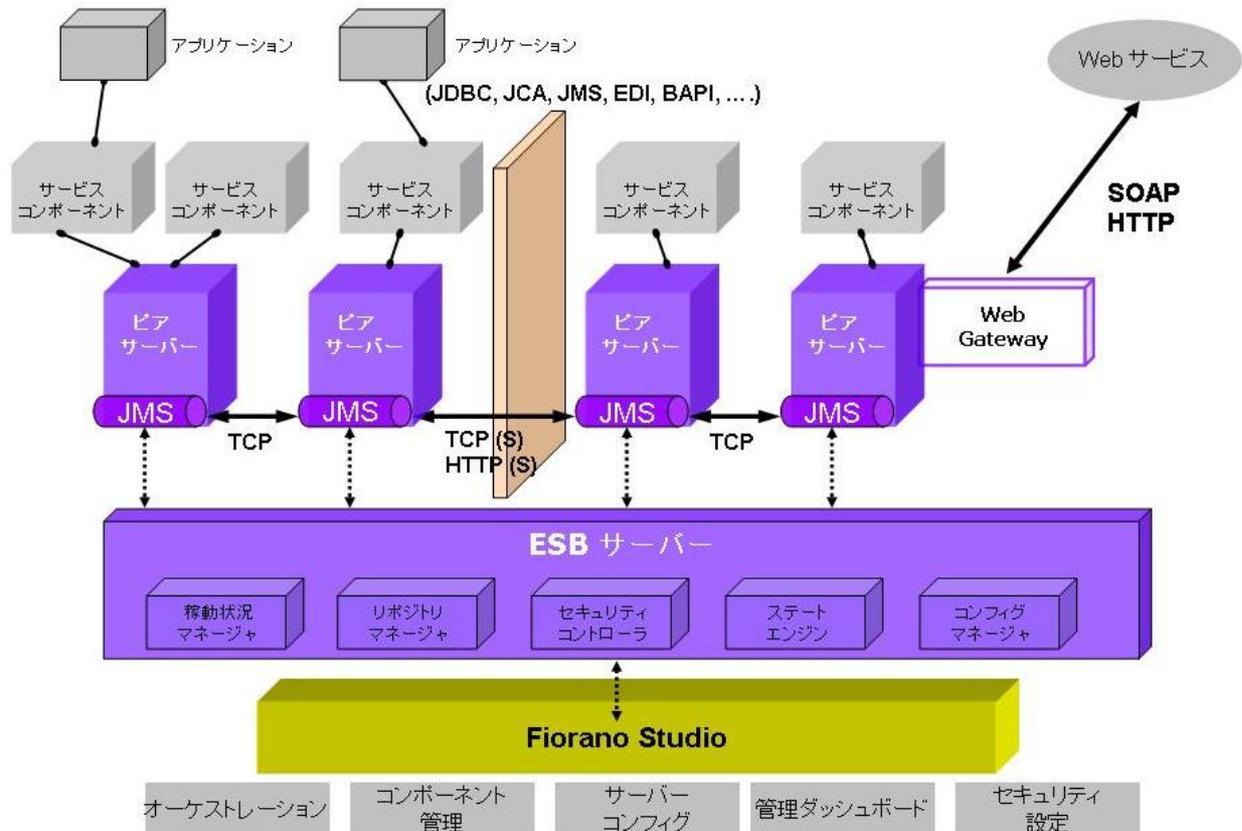
上図は、SOAP over HTTP を介した Web サービスの呼び出しに、WebGateway サーバーを使用している様子を示しています。

WebGateway は、Web サービスの呼び出し以外にも、以下の機能にも利用されます。

- Fiorano ESB 上のビジネス プロセスを Web サービスとして公開
- Fiorano のワークリスト管理機能のホスティング、およびワークリスト web ページとの間の転送 (ワークリストとは人間による処理をビジネス プロセスに組み込むためのメカニズムで、Fiorano SOA にはワークリストを実現するためのサービス コンポーネントが備わっています。)

1.3 Fiorano SOA プラットフォームの全体アーキテクチャ

下図は、Fiorano SOA プラットフォーム の全体的なアーキテクチャを図示しています。



各サーバー (ピア、ESB サーバー) は、その実行 OS を選びません (JRE 1.5 以降が必要です)。このため、様々なマシンや OS が混在した環境においても、1 本の ESB として機能することができ、異なる種類のマシン上で稼働しているピアサーバーやビジネス プロセスを一元的に管理できるようになっています。

上図には、今までに説明していない“サービス コンポーネント”が図示されています。サービス コンポーネントは、ビジネス プロセスを構成するためのコンポーネントです。

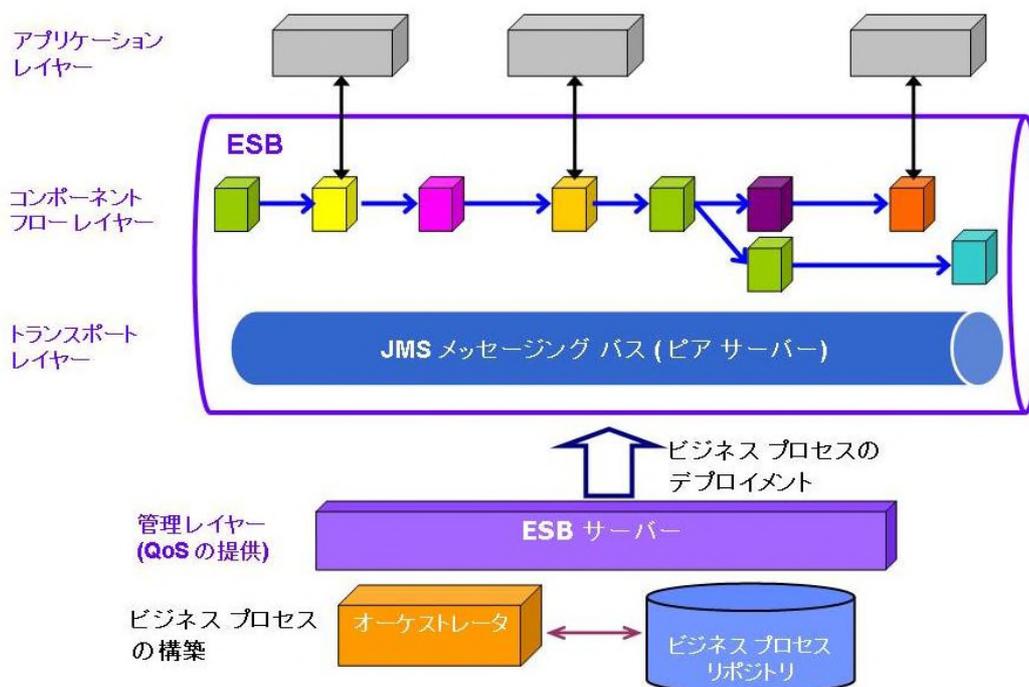
詳細は、次章『2 ビジネス プロセスの概要』で説明します。

2 ビジネス プロセスの概要

2.1 ビジネス プロセスのレイヤー構造

Fiorano SOA プラットフォームは、SOA 実装フレームワーク (SOAIF : SOA Implementation Framework) に基づいたビジネス プロセスを構築、実行します。SOAIF とは Fiorano 社の SOA コンセプトを実現するためにデザインされた、ビジネス プロセスを実装するためのフレームワークです。SOAIF によって実装されるビジネス プロセスは、下図に示すレイヤー構造を採っています。

- **アプリケーション レイヤー** (アプリケーションのビジネス ロジックを提供するレイヤーで、SOA のサービスに相当する)
- **コンポーネント フロー レイヤー** (サービス コンポーネントを結合させることで構成するプロセス フロー。ビジネス プロセスのフロー制御を担い、アプリケーション レイヤーが提供するビジネス ロジックの呼出し順序の制御やアプリケーション ロジック間のメディエーション機能を実現する)
- **トランスポート レイヤー** (ピア サーバーによって構成されるメッセージ (データ) 伝送路)
- **管理レイヤー** (ビジネス プロセスの開発環境を提供するとともに、プラットフォームの QoS (クオリティ オブ サービス) を担う)



アプリケーション レイヤー

実際のアプリケーションが稼動するレイヤーで、ビジネス ロジックを実行します。Fiorano のビジネス プロセスでは、アプリケーションに対して特別な設定やモジュール追加などを行う必要がありません。また、Web サービスのインタフェースを備える必要もありません。このため、既存アプリケーションを現行の運用状態のまま、Fiorano のビジネス プロセスの中に組み込むことができます。

各アプリケーションは、コンポーネント レイヤーのコンポーネントから呼び出されることで、ビジネス プロセスにおけるビジネス ロジックを実行します。このアプリケーション呼び出しは、アプリケーションの呼び出しプロトコルに応じた専用のサービス コンポーネントによって行われます。Fiorano SOA プラットフォームには、多種多様なプロトコルに対応したコンポーネントをあらかじめプリビルトして製品にバンドルしています。このため、ほとんどプログラミングすることなく、多くのアプリケーションを

ビジネス プロセスの中に組み込むことができます。

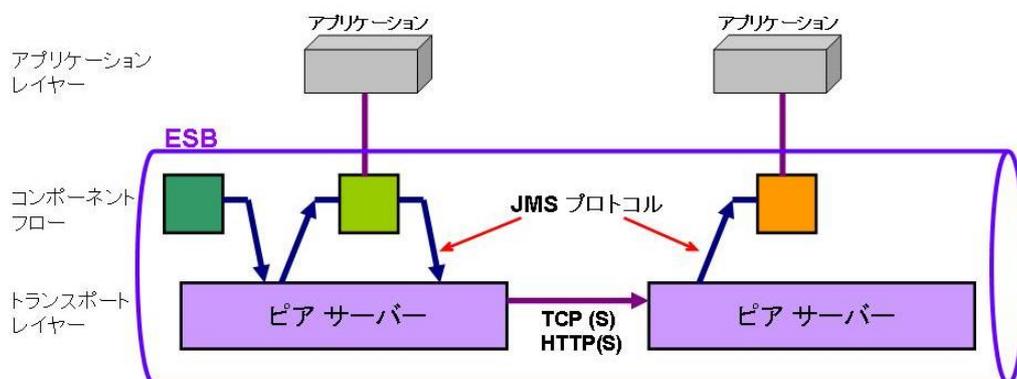
コンポーネント フロー レイヤー

ビジネス プロセスのフロー制御を担っているのがコンポーネント フロー レイヤーで、ビジネス プロセスのフローを具現化したものとなります。Fiorano SOA プラットフォームにおけるビジネス プロセスは、コンポーネント フロー レイヤーとアプリケーション レイヤーによって構成されます。コンポーネント フローは、サービス コンポーネントを組み合わせて、結合していくことで構築します。各コンポーネントは、実際に移動するソフトウェアであり、モジュラー技術の利点をフルに生かしたものです。コンポーネント フローの構築は、オーケストレーション ツール（Fiorano Studio の中から使用します）を用いてグラフィカルに、またほとんどマウス操作のみで行えます。Fiorano SOA プラットフォームには、プリビルトされて製品にバンドルされているサービス コンポーネントが 60 種類以上あり、ビジネス プロセスのすべてをユーザーが開発する必要がありません。

このレイヤーは、ビジネス プロセスを抽象化してユーザーに提示します。実際のアプリケーションの稼働環境やトランスポート レイヤーをユーザーから隠蔽することができるため、IT システムに詳しくないビジネス部門のユーザーでもビジネス プロセスを構築することが可能となります。

トランスポート レイヤー

メッセージ（データ）の実際の伝送路を形成するもので、ピア サーバーがメッセージングのエンジンとして機能します。コンポーネント フローは、オーケストレーション ツールを用いて開発され、ESB サーバー上のリポジトリに格納されます。ビジネス プロセスの実行時に、ESB サーバーから関連する各ピア サーバーに必要なリソース（サービス コンポーネント、コンポーネント フローの定義ファイルなど）がデプロイメントされます。デプロイメントされた個々のサービス コンポーネントは、ピア サーバーの JMS に完全に準拠したメッセージング機能を介して他のサービス コンポーネントとの間でメッセージ（データ）の送受信を行います。



各コンポーネント間の統合エンジンとして、ピア サーバーは JMS サーバーとしての機能を果たします。これによって、コンポーネント フローは、メッセージングをベースとしたイベント ドリブなビジネス プロセスとして実行されることとなります。また、次にあげるような JMS のセマンティックをフルに活用することができます。

- ストア & フォワードによるメッセージ配信の保証
- 恒久サブスクリプション、メッセージの永続化
- Acknowledge、トランザクション モード
- パブリッシュ / サブライブ方式による複数の送信先へのメッセージ配信

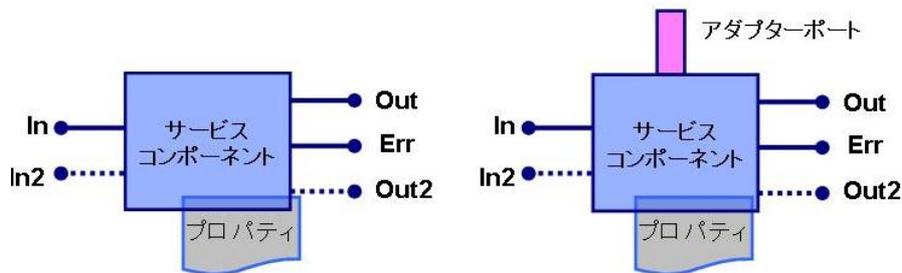
管理レイヤー

ビジネス プロセスの実行基盤である ESB は、ESB サーバーによって集中管理されます。ESB のメッセージング バスが分散された各拠点全体にまたがって配置されるのとは異なり、ESB サーバーは任意の拠点一箇所に配置され、ESB ネットワークの全体を集中的に管理します。ESB サーバーには、ESB ネットワーク全体の QoS (クオリティ オブ サービス) を高めるために、多彩な管理ツール (ロギング、監視、セキュリティ設定、フォールトトレランス機能など) が用意されています。

2.2 サービス コンポーネント

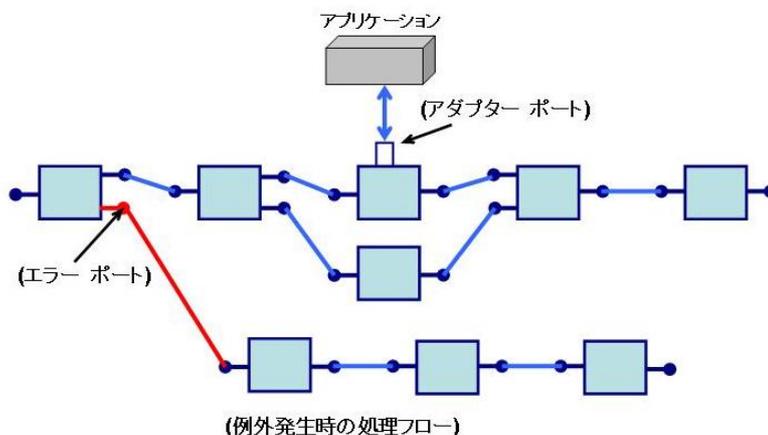
サービス コンポーネントは、コンポーネント フローを形成するビルディング ブロックです。個々のサービス コンポーネントは、特定のタスクを他のサービス コンポーネントとは独立して単独で実行する単体のアプリケーション ソフトウェアです。

サービス コンポーネントには、メッセージ (データ) 送受信するためのインプット ポートとアウトプット ポートが備わっています。インプット ポートおよびアウトプット ポートは、それぞれ複数持つことができます。また、通常のメッセージ (データ) 送受信ではなく、例外発生時のエラー メッセージを送信するためのエラー ポートも 1 つ備えています。アプリケーション レイヤーにあるアプリケーションを呼び出すためのアダプター ポートを備えたサービス コンポーネントもあります。



サービス コンポーネントのアウトプット ポートを別のサービス コンポーネントのインプット ポートへ繋げることで“疎結合”な連携を実現します。下の図は、ポート間の連携の様子を示しています。アウトプット ポートを複数設けることによるフローの分岐、エラー処理フロー、アプリケーション呼び出しなどのイメージが理解いただけるものと思います。(フロー制御に特化したサービス コンポーネントも用意されており、より複雑なフロー制御が行えるようになっています。)

図ではピア サーバーを省略していますが、実際にはコンポーネント間の結合はピア サーバーを介した JMS メッセージングによって実現されています。



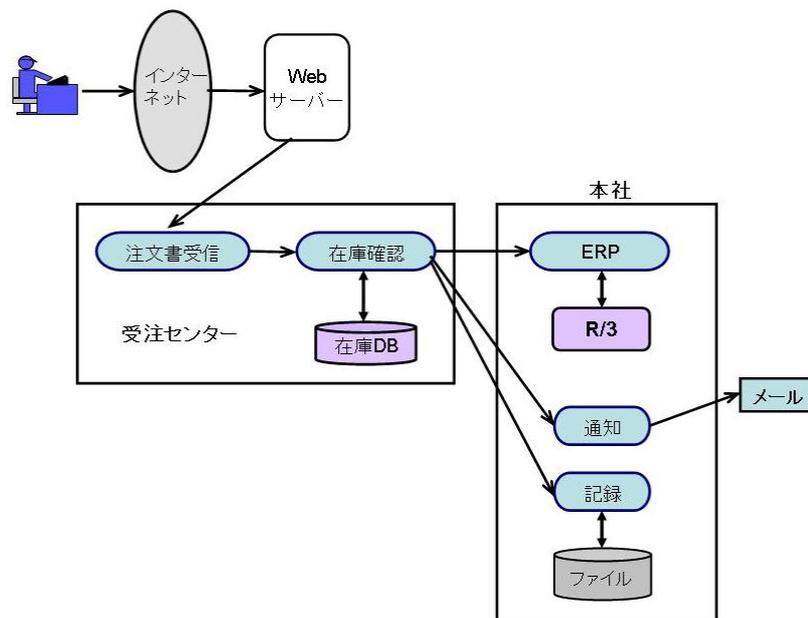
アダプター ポートは、呼び出すアプリケーションに応じたプロトコルを使用します。例えば、RDBMS を呼び出すサービス コ

コンポーネントのアダプター ポートは JDBC を使用し、SAP R/3 システムを呼び出すサービス コンポーネントは、BAPI / RFC または IDOC / ALE を使用します。

Fiorano SOA プラットフォームには、Fiorano 社によってプリビルトされたサービス コンポーネント (60 種以上)と、ユーザー独自のコンポーネント開発のための開発キットが付属しています。

2.3 ビジネス プロセス ビューと物理的なデプロイメント

次の図に示すような簡単でシンプルな受注システムを例題にして、Fiorano SOA プラットフォームにおけるビジネス プロセスの構築とデプロイメントについて説明します。



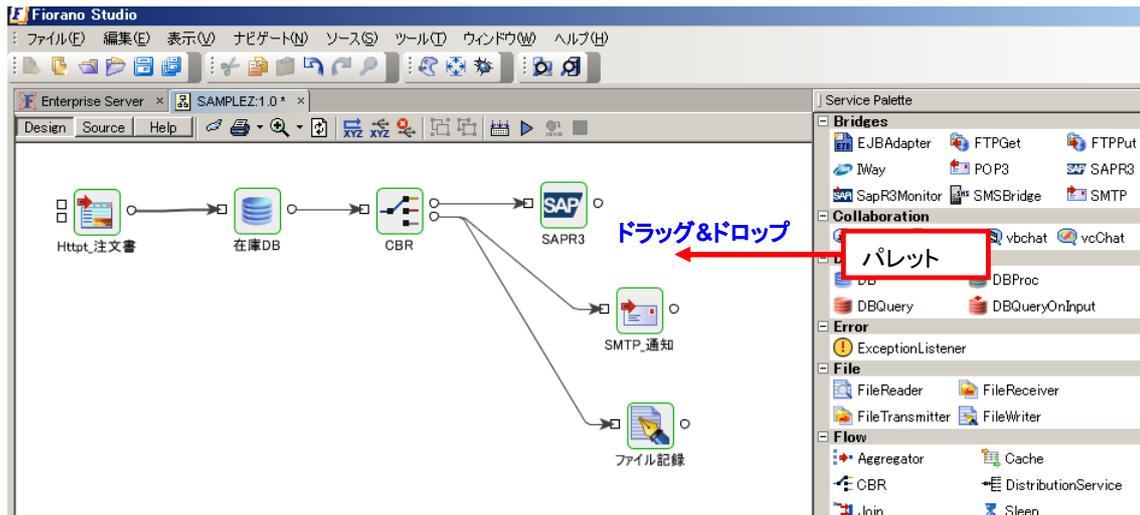
図に示した受注システムは、次の業務処理ステップによって注文書の処理を行います。

1. ユーザーからの注文書を受信 (ユーザーはポータル サイトから注文書を発行します)
2. 注文のあった製品の在庫確認 (在庫確認は在庫データベースに直接アクセスして行います)
3. 在庫がある場合には ERP (SAP R/3) にある ABAP プログラムの実行を依頼し、受注処理を行います
4. 在庫がない場合には、担当者に電子メールで通知します
5. 同時に、在庫切れとなった状況をファイルに記録します

また、図に示すように、在庫データベースは受注センターで稼働しており、ERP は本社で稼働しています。在庫切れの記録は、本社のファイルに記録します。

ビジネス ビュー

次の図は、オーケストレータ ツールによって作成したコンポーネント フローをキャプチャしたものです。



このコンポーネント フローは、Fiorano がプリビルトして提供しているコンポーネントのみで構成されています。コンポーネント フローの作成は非常に簡単で、次の作業を行うだけです。

1. パレットから必要なコンポーネントをドラッグ & ドロップ
2. コンポーネント間をマウスで結びつける

コンポーネント フローでは、受注センターで稼働しているアプリケーションも、本社で稼働しているアプリケーションも 1つのプロセス フロー上に表すことができ、企業内外にまたがる業務プロセスを統括してデザインすることができます。

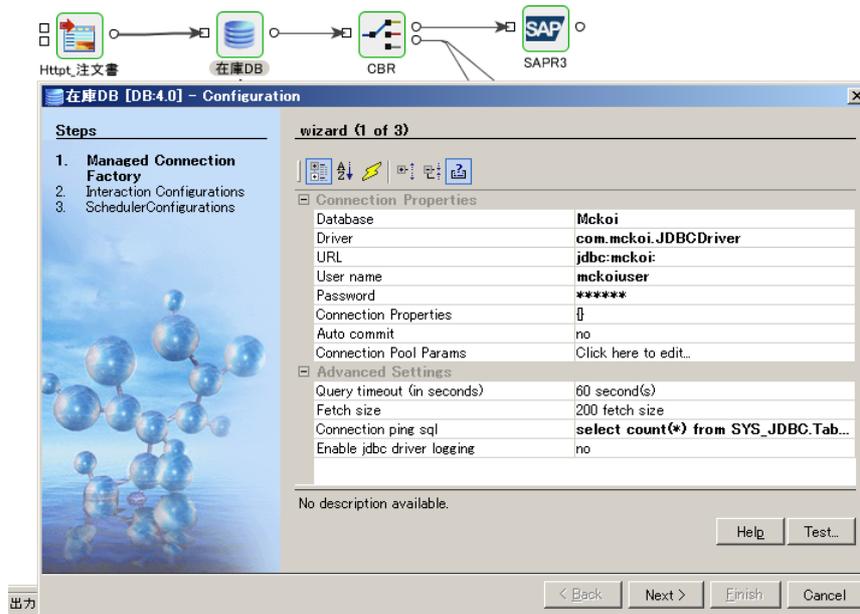
また、このフロー図には、ネットワーク構成やハードウェア構成、サーバー設定といった IT 技術の専門知識を必要とするような項目は表示されていません。

コンポーネント フローは、IT 技術に詳しくない業務担当者でも作成可能であり、ビジネス担当者が業務プロセスを設計、検討するためのビューを提供することができます。

物理的な IT 環境の定義

コンポーネント フローを実行するためには、実際の稼働環境に即した定義が必要となります。

次のキャプチャ画面は、在庫データベースへのアクセスを定義している様子を示しています。

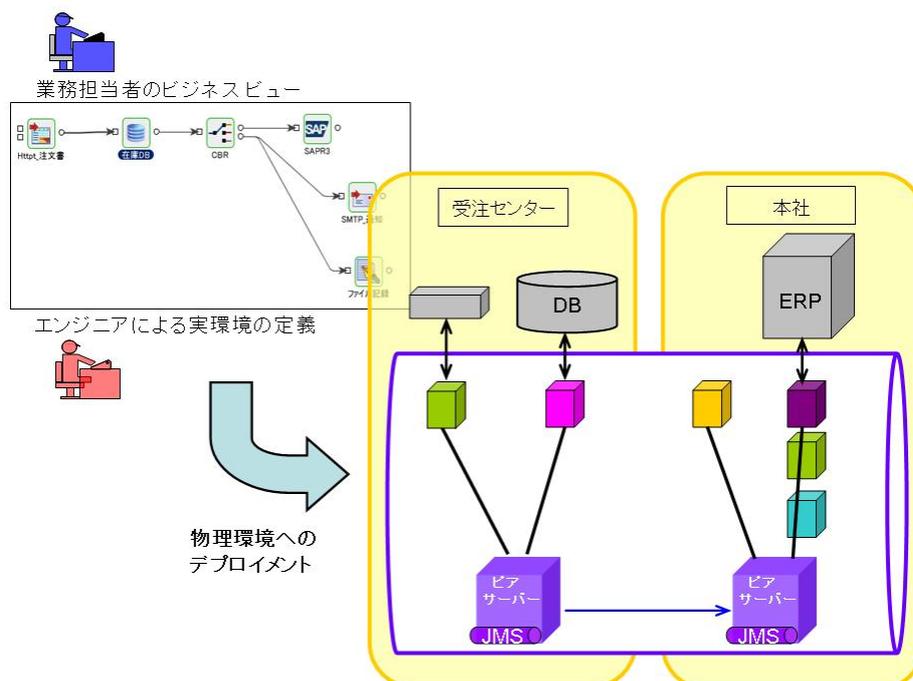


コンポーネント フローの実環境への適用は、個々のコンポーネントのプロパティ設定によって行います。上の画面は、在庫データベースにアクセスする JDBC ドライバーを定義するプロパティ シートです。プロパティの定義には IT に関する技術知識が必要であり、通常、IT エンジニアによって行われます。

コンポーネント フローは、業務短担当者へのビジネス ビューを提供すると同時に、IT エンジニアに対する実行環境の定義ビューも提供します。それぞれの得意とする分野のデザインや定義を同一のコンポーネント フローに対して行えることは、ビジネス プロセスのモデリングと実装の両方において大きな生産性の向上をもたらします。

実環境へのデプロイメント

オーケストレータ ツールで作成したコンポーネント フローは、次の図に示すように、実環境にデプロイされ、稼動します。



3 プリビルト コンポーネントの一覧

Fiorano SOA プラットフォームには、Fiorano 社によって開発されたサービス コンポーネントが製品にバンドルされています。このプリビルト コンポーネントを利用することで、コンポーネント フローを無から作成する必要がなくなり、開発期間を飛躍的に短縮できます。

ブリッジ タイプ (外部アプリケーションの呼び出し、外部システムとのデータ送受信)

カテゴリ	コンポーネント	アダプター ポートのプロトコル
EJB ブリッジ	EJBAdapter	J2EE
FTP ブリッジ	FTPGet、FTPPut	FTP
メール ブリッジ	POP3、SMTP SMSBridge (携帯電話ショーとメッセージの送信)	メール プロトコル SMS プロトコル
iWay 社アダプター	IWay (iWay 社アダプターへのリクエスト送信)	JCA
RDBMS	DB (SQL クエリ (ネスト可) の実行、ストアード プロシージャ コール、テーブル変更のモニタリング) DBProc (ストアード プロシージャ コール) DBQuery (単一の SQL クエリの実行) DBQueryOnInput (インプット ポートから受信した SQL クエリを実行)	JDBC
エクセル ファイル	DB	ODBC
HL7	HL7Receiver、HL7Sender (医療標準データの送受信)	HL7 lower プロトコル
MOM	JMSIn、JMSOut、JMSReplier、JMSRequestor MQSeriesIn、MQSeriesOut MSMQReceiver、MSMQSender TibcoRVIn、TibcoRVOut	JMS IBM MQSeries Microsoft MQ Tibco Rendezvous
SAP R/3	SAPR3 (BAPI/RFC による R/3 へのリクエスト送信) SAPR3Monitor (IDOC による R/3 データ変更の受信)	BAPI/RFC IDOC/ALE
Web	HttpAdapters (Web サーバーとの間の HTTP Get/Post リクエストの送信) HttpReceiver (Web サーバーとの間の HTTP Get/Post リクエストの受信およびリプライの返信) SimpleHTTP (ベーシックな GET / POST リクエストの送信)	HTTP
Web サービス	WebServiceConsumer (Web サービス呼び出し) WSStub (ビジネス プロセスを Web サービスとして公開するためのコンポーネント)	SOAP over HTTP ビジネス プロセスの Web サービス化

ファイル アダプター

カテゴリ	コンポーネント
ファイル読み込み	FileReader (ファイルの内容を読み込む)
ファイル書き込み	FileWriter (ファイルにデータを書き込む)
大容量ファイル	File Transmitter ファイルの受信 (File Receiver と合わせて大容量ファイルの転送に使用) FileReceiver (ファイルの受信 ((File Transmitter と合わせて大容量ファイルの転送に使用)

フロー制御

カテゴリ	コンポーネント
コンテンツ ベース ルーティング	CBR (XPath シンタックスに従ったルール エンジンとして機能。XPath で指定した内容を持つ XML データを識別し、指定されたアウトポート ポートに振り分ける)
結合	Join (2つの経路から送信されてきた XML を 1つの XML に結合する)
合成	Aggregator (複数の経路から入力した XML を 1つの XML に結合する)
負荷分散	DistributionService (メッセージの処理を同一の処理を行う複数のフローへ分散させる)
時間制御	Sleep、Timer
エラー処理	ExceptionHandler (コンポーネント フロー中で発生したエラーを一元的に集中処理する)

データ サービス

カテゴリ	コンポーネント
データ フォーマット変換	Text2XML、XML2Text (ユーザー指定のスキーマに基づく変換) XML2PDF (ユーザー指定のテンプレートに基づいて、XFDF フォーマットの XML を PDF ファイルに変換) EDI2XML、XML2EDI (EDI フォーマットと XML との間の変換。EDI フォーマットの事前定義が必要) Xslt (XSLT に基づく XML から XML へのデータ変換)
PDF 変換	XML2PDF (XML 形式のデータを PDF に変換 (変換テンプレートを事前に定義))
圧縮	Compression、Decompression
暗号化	Encryption、Decryption (DES による暗号化)
XML の分割	XMLSplitter (XPath シンタックスに従った XML の分割)
XML の検証	XMLVerification (XSD に基づく XML データの内容チェック)

スクリプト

(ユーザー独自のデータ変換ロジックをスクリプト言語を用いて記述することができるコンポーネントです。コンポーネントの入力ポートから受信したデータをスクリプト言語によって記述されたロジックで変換し、アウトポート へ送信します。)

カテゴリ	コンポーネント
Bean Shell	BeanShell (インプット メッセージを記述されたスクリプトで処理します)
Java Script	Javascript (インプット メッセージを記述されたスクリプトで処理します)
Perl	PerlScript (インプット メッセージを記述されたスクリプトで処理します)
Python	Python (インプット メッセージを記述されたスクリプトで処理します)
Groovy	GroovyScript (インプット メッセージを記述されたスクリプトで処理します)

ワークリスト (人間による処理の介入...承認フローに利用)

カテゴリ	コンポーネント
ワークリストへのタスク追加 (担当者の作業リストへの作業の追加および次アクションの受信)	<p>Work List</p> <p>指定された担当者に作業を通知 (通知は web サーバー上に記録)</p> <p>担当者は、作業通知を Web ブラウザを用いて確認し、次ステップとしてのアクションを選択する。選択されたアクションが Work List コンポーネントに返され、それに応じたアウトポート ポートから送信される。(アウトポート ポートを分けることで次の処理フローに振り分けられる。)</p>
システム用コンポーネント	<p>Work List Manager</p> <p>Work List コンポーネントと Web サーバー上の web ページとの間を仲介するコンポーネント。システムが自動的に起動するので、ユーザーが明示的に用いる必要はない。</p>