



IBM Software Group

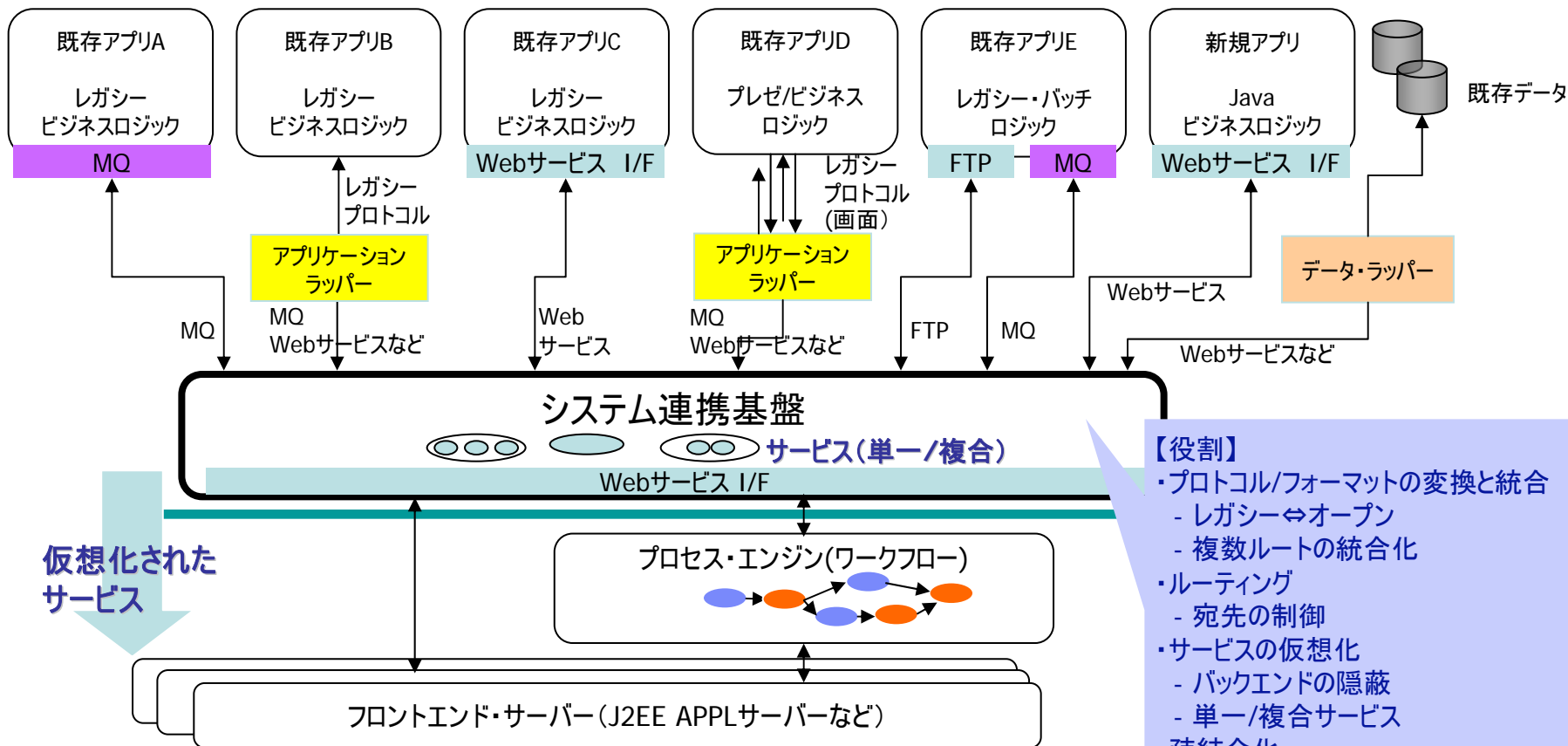
WebSphere Message Brokerで実現する 次世代オンラインシステム連携基盤

目次

- システム連携基盤とは？
- WebSphere Message Brokerのご紹介
- システム連携基盤の5つのポイント
 - ① 既存資産の最大活用
 - ② 開発生産性と変更容易性
 - ③ パフォーマンス
 - ④ 高信頼性・高可用性
 - ⑤ セキュリティ

システム連携基盤とは

システム連携時に使用できるのはオープンなインターフェースだけではありません。
 既存資産の中にはオープン化が難しいもの/オープン化するためのコストがかかるものがあります。
 既存インターフェースをシステム連携基盤にそのまま連携できれば、追加開発/メンテナンスコスト削減につながります。



- 【役割】**
- ・プロトコル/フォーマットの変換と統合
 - レガシー⇄オープン
 - 複数ルート of 統合化
 - ・ルーティング
 - 宛先の制御
 - ・サービスの仮想化
 - バックエンドの隠蔽
 - 単一/複合サービス
 - ・疎結合化
 - システム/アプリの追加・変更の容易化、影響の局所化

など



システム連携基盤の5つのポイント

システム連携基盤を提案・構築・展開していくにあたり、ポイントとなる項目を挙げます。

- ① 既存資産の最大活用
- ② 開発生産性と変更容易性
- ③ パフォーマンス要件
- ④ 高信頼性・高可用性
- ⑤ セキュリティ



IBMのESB製品

WebSphere delivers the most complete ESB solution



WebSphere ESB

WebSphere Application Serverベースの統合SOA環境

- ・WAS NDベースのESBソリューション
- ・Webサービスとの高い親和性
- ・WASの運用管理方法などを継承

ESB offerings from IBM WebSphere



WebSphere Message Broker

異機種間での広範な接続インターフェースと変換を実現

- ・WebSphere MQベースのESBソリューション
- ・広範な接続性、データ変換をサポート
- ・既存システムとの連携、WebSphere MQとの高い親和性
- ・高スループットが要求される業務に適合
- ・基幹業務での豊富な実績



WebSphere DataPower Integration Appliance

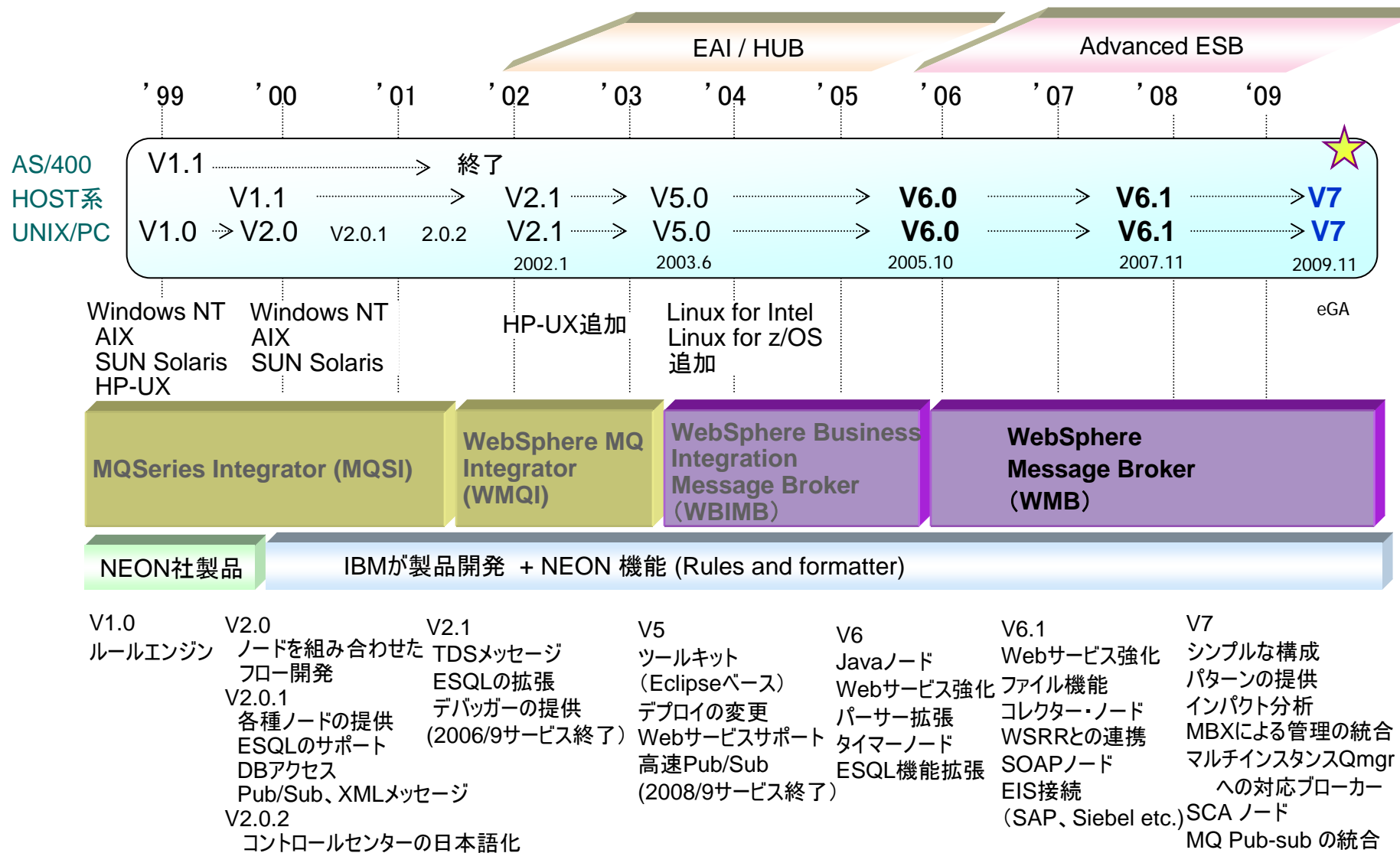
シンプルな操作と強固なセキュリティを提供するハードウェアESB

- ・SOAアプライアンスによるシンプルな構成
- ・高速なXML処理

本資料では

この製品に焦点を当てます。

WebSphere Message Brokerの歩み



WebSphere Message Broker (WMB)の強み

WebSphere Message Broker (WMB)は幅広いトランスポートプロトコルのサポートと、強力なパーサーによる幅広いフォーマットサポートを活かした、強力な変換機能を提供します。多くの機能を既製部品として提供しており、開発工数低減に寄与する一方、柔軟なカスタマイズも可能です。



WMB

1. 強力な変換機能の提供
 - 幅広いトランスポートプロトコルのサポートと、それら間の変換の提供
 - 幅広いフォーマットのサポートとそれら間の変換の提供
2. プロトコル変換、フォーマット変換といったESB主要機能を既製部品の組み合わせで実装し、極力コーディングを排除
3. 柔軟なカスタマイズ機能 (Plug-inノード, パーサー, 文字コードテーブル, EXIT)
4. WMQをベースとした高スループット・高信頼性が要求される処理への適合性。トレース機能、デバッグ機能といった問題判別機能の充実。
5. XML, WSDL, SOAPといった標準技術への準拠 (WS-I basic profile 1.1への準拠など)
6. 金融機関を中心とする基幹業務での豊富な稼働実績

エントリー・オプションについて

2008年4月2日付けで、UNIX/PC系のWMBに新たにエントリー・オプションが追加されました。

<http://www-06.ibm.com/jp/domino02/NewAIS/aisextr.nsf/ByLetterNo/SWA08019?OpenDocument&ExpandSection=1&highlight=0,%5BTitle0%5D=WebSphere,Message,Broker>

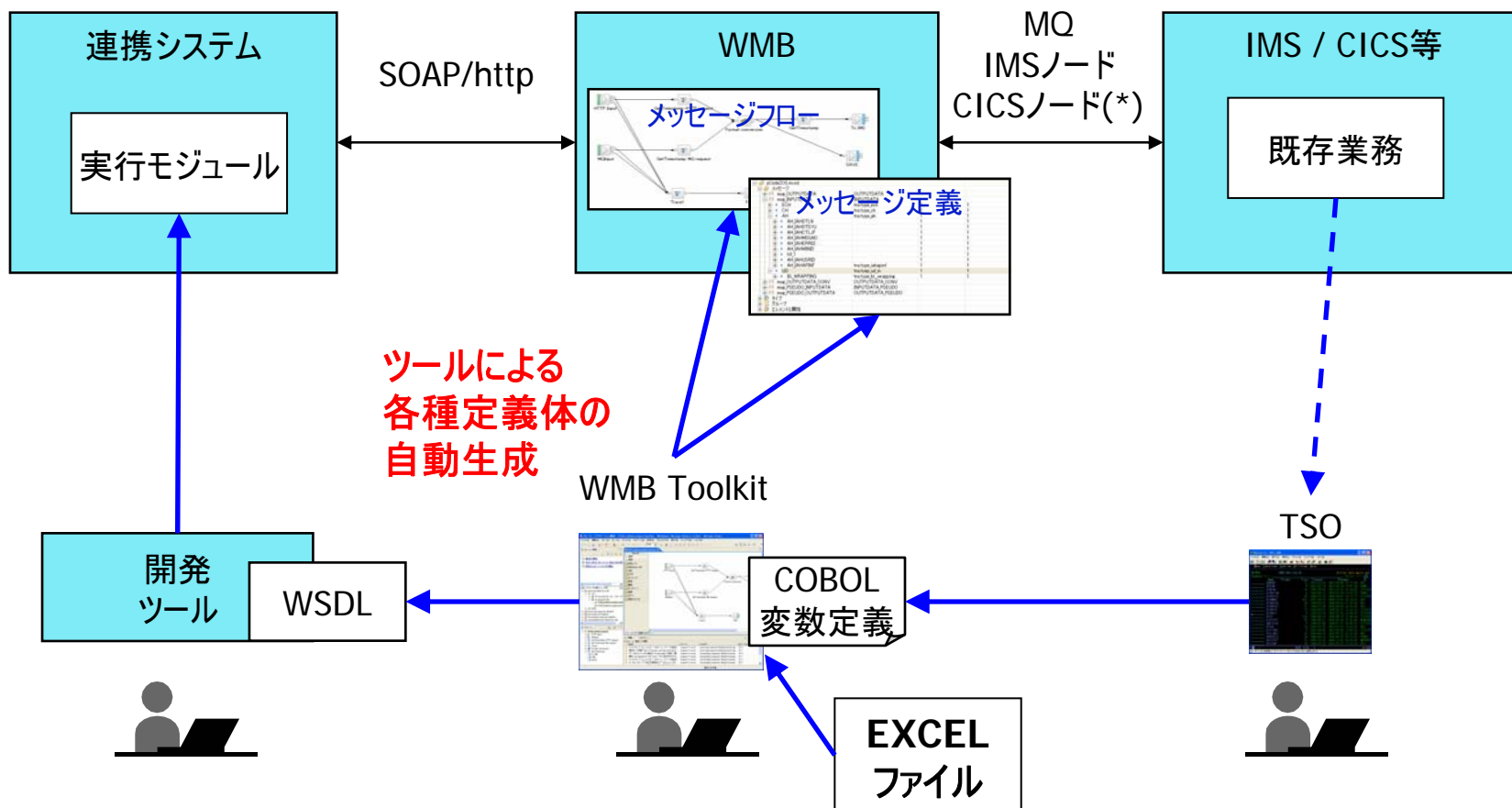
- WebSphere Message Broker Starter Edition
 - Small Start用の安価なライセンスです。
 - 導入できる実行グループは1つ、メッセージ・フローは 10 個までという制限があります。
 - WebSphere Message Broker を導入するサーバーは、ユニプロセッサまたはデュアルコア・プロセッサに限られます。
 - WebSphere Message Broker Starter Edition は、無制限バージョンにアップグレード可能です。

- WebSphere Message Broker for Remote Adapter Deployment
 - WebSphere Message Broker を使用して WebSphereアダプターを利用することに特化したライセンスです。
 - WebSphere Message Broker for Remote Adapter Deployment には、以下の制限があります。
 - 利用できるノードに制限があります(アダプター・ノード、Java Computeノード、入出カトランスポート・ノード)。
 - アプリケーション・インスタンス単位および設定単位の標準的な WebSphere Adapter ライセンス交付ルールが適用されます。
 - 導入できる実行グループは 2 つまでに制限されています。
 - WebSphere Message Broker およびアダプターを導入するサーバーは、ユニプロセッサまたはデュアルコア・プロセッサに限られます。

- WebSphere Message Broker Trial Edition
 - WebSphere Message Broker については、90 日間のトライアル・ダウンロード版を入手可能です。
 - 詳細は以下のサイトにアクセスしてください。
<http://www-306.ibm.com/software/integration/wbimessagebroker/>

IMS・CICSとの容易な接続

- ① 既存資産の最大活用
- ② 開発生産性と変更容易性



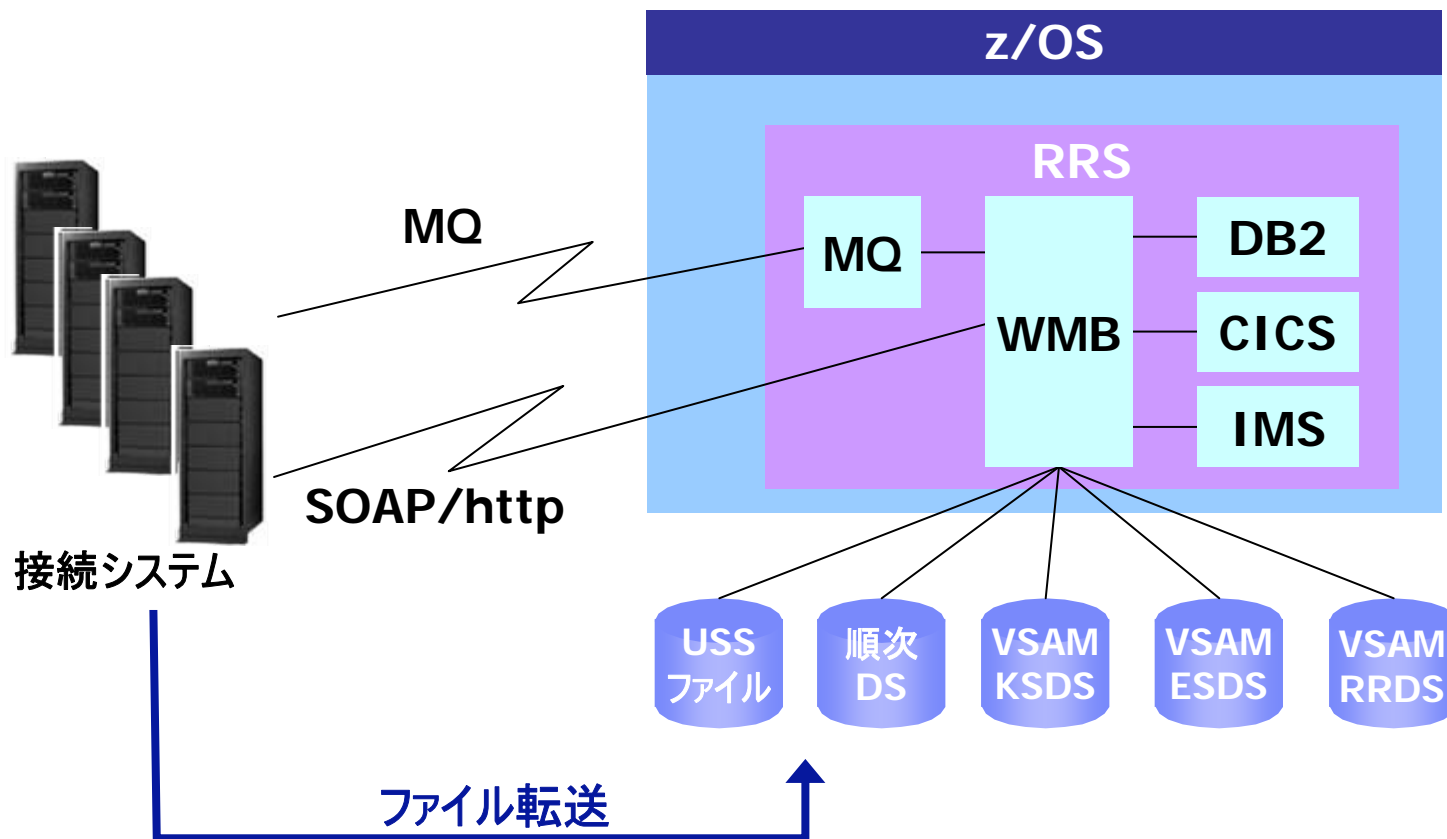
- 既存資産を「容易」にサービスとして活用できるか否かは連携基盤利用促進においてとても重要です

(*)CICSノードはWMB for z/OS版のみの機能です。

既存資産活用について(WMB for z/OS)

WMB for z/OSはメインフレームの既存資産とのインターフェースを標準で装備しています。既存業務をうまく活用した次のような利用方法が考えられます。

- CICS、IMSアプリケーションの問い合わせ結果を集約して返答
- FTPで送信されたファイルを入力にオンラインアプリケーションをバッチ的に処理
- VSAMファイルのデータ内容をWebサービスで公開



開発生産性 – 例としてMQInputノード

WMBの入出インターフェース実装は既存部品(プロセスノード)にプロパティをセットするのみです。接続インターフェースの進歩に独自開発で追隨していくのは時間もコストもかかります。接続機能のサポートは、ミドルウェアを活用する価値のある箇所です。

MQInput ノード・プロパティ - MQInput

キュー名* **必須はこれだけ**

MQInput ノード・プロパティ - MQInput

ID トークン・タイプ

ID トークンのロケーション

ID パスワードのロケーション

ID issuedBy のロケーション

セキュリティ例外を通常の例外として処理します

構文解析のタイミング

MQRFH2C コンパクト・パーサーを MQRFH2 ヘッダーに使用

XMLNSC パーサー・オプション

XML スキーマ・データ・タイプを使用したツリーのビルド

XMLNSC コンパクト・パーサーを XMLNSC ドメインに使用

混合内容の保存

コメントの保存

処理命令の保存

不透明なエレメント

エレメント	追加...	編集...

MQInput ノード・プロパティ - MQInput

妥当性検査

失敗時の処理

MQInput ノード・プロパティ - MQInput

メッセージ・ドメイン

メッセージ・セット

メッセージ・タイプ

メッセージ形式

トランザクション・モード

順序モード

論理順序

すべてのメッセージが使用可能

一致するメッセージ ID

一致する相関 ID

変換

エンコードの変換

コード化文字セット ID の変換

メッセージ・グループ別にコミット

z/OS シリアライゼーション・トークン

トピック

開発生産性の評価例

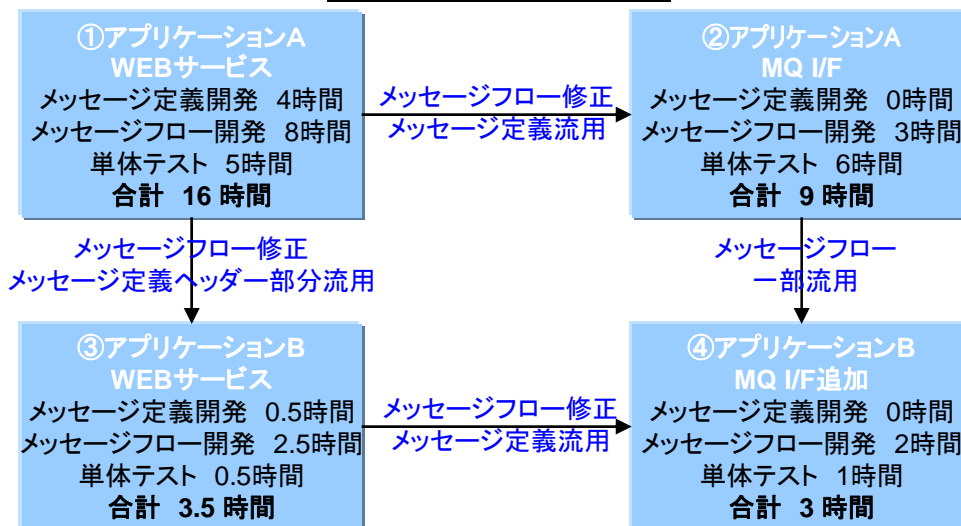
開発生産性について

- 今回WMBを使用したテストケースは4タイプありましたが、右図の①～④の順に開発いたしました。
- メッセージ定義はテストケース①において、COBOLコピー句を元に作成しましたが、バイナリーフィールドのカスタマイズ、WSDL生成等に4時間要しました。メッセージ定義の大部分はヘッダーでしたので、残りのテストケースではそのまま流用することによって、工数はほとんどかかかっていません。
- メッセージフローについては、テストケース①で作成したものをベースにする事によって、後続のテストケースでの開発工数は大幅に低減できました。
- 今回WMBはz/OS版を使用しましたが、単体テストはWindows版で行いました。また、WPS、IMS等の連携システムをエミュレートする処理もメッセージフローとして開発する事によりWMB内でテストを完結させることとなり、単体テストの作業効率、妥当性を高めています。

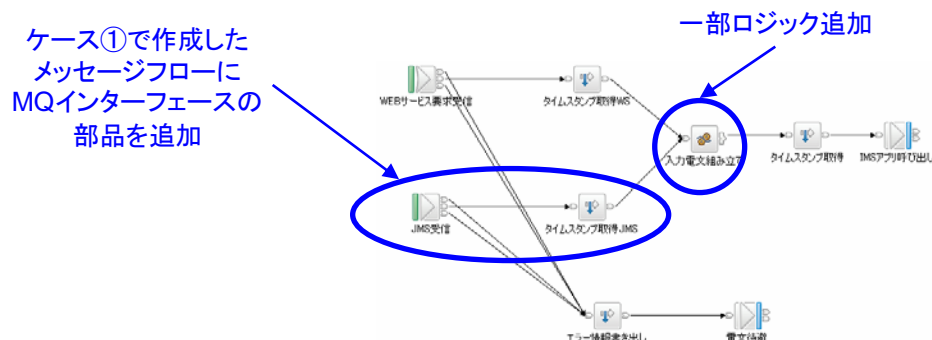
変更容易性について

- ヘッダー部等の多くの処理で共用されるものは、共通部品として定義して、各メッセージ定義で共用できます。メッセージフローも共通部分をサブフローとして定義して、各メッセージフローからの共用が可能です。
- キュー名や、WEBサービスURL、データソース名といった環境によって変化するプロパティは、メッセージフローそのものを変更しなくても、barファイル作成時に変更できます。外部から値の指定が可能なユーザー独自の変数を定義する事も可能です。
- チーム開発では、メッセージ定義、メッセージフローの定義体はCVS等のバージョン管理ソフトウェアで管理する事を推奨しています。

開発生産性の評価



既存メッセージフローの流用



③	パフォーマンス
④	高信頼性・高可用性
⑤	セキュリティ

システム連携基盤の非機能要件

一般に複数のシステムの仲介を行う連携基盤は、求められる非機能要件レベルが高い事が多いです。

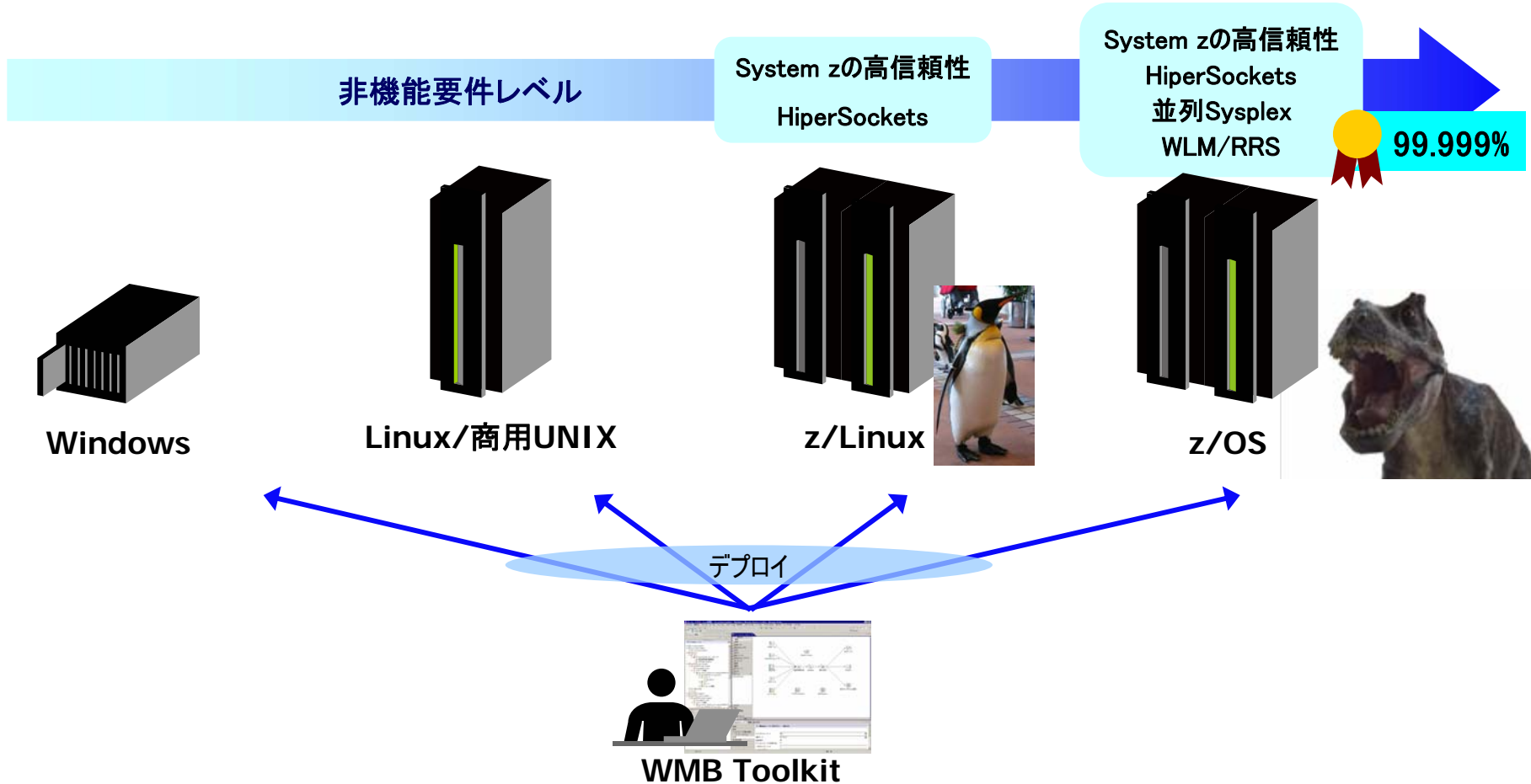
- 高パフォーマンス、高スループット
 - 秒100件以上のスループット要件
 - 厳しい応答時間要件
- 信頼性
 - 堅牢性
 - 計画外停止ゼロの追求
 - 連続稼働
 - 連携システム数の増加、トランザクション量の増加にダイナミックに対応
 - 高可用性
 - メッセージの消失防止
- セキュリティ
 - 監査証跡
 - メッセージの改ざん防止、情報漏えい防止



- ③ パフォーマンス
- ④ 高信頼性・高可用性
- ⑤ セキュリティ

WMBのプラットフォームについて

WMBの開発物(メッセージセット、メッセージフロー)は、どのプラットフォームでも等しく稼働するようにデザインされています(ファイルI/O、文字コード等のプラットフォーム依存部分を除く)。WMBのプラットフォーム選定は、どれだけの非機能要件レベルを満たす必要があるかによって行います。



高信頼性・高可用性を実現するSystem z10

IBMメインフレーム(System zやz/Linux)の信頼性は、既に多くのお客様で実証済

■ System zの特徴

連続稼動を実現する技術

ダウンタイムゼロを追求するSystem z
ミッションクリティカルな業務での豊富な稼動実績

実績を伴った仮想化品質

40年以上の仮想化品質
System zでは仮想化はあたりまえ
H/WだけでもCPU、ネットワーク、ストレージの共用が可能
多くのLinux区画を実装できるスケーラビリティ

システム全体のスケーラビリティ

単体の性能ではなくシステム全体の性能を追求



System z

全てを兼ね備えた超ハイブリッド・仮想化サーバー

WMB構成のポイント

③ パフォーマンス

④ 高信頼性・高可用性

WMB構成のポイントを記述します。中でもデータベースの用途と配置は大きな要素となります。

■ ハイトラフィックへの対応

- WMBはスケールアップ、スケールアウト共に良好な適正を持っている
- **ユーザーDBへのアクセスが必要な場合、DBアクセスがボトルネックとならないような設計が肝要**
- スケールアウトについては負荷分散装置をフロントに配置する事が基本。MQのキューマネージャーとの通信時はMQクラスターを利用。

■ データベースの配置

- WMBと同じOSに配置
 - パフォーマンスが良い
 - ユーザーDBを配置する場合、OS数によってはライセンス費用が増大(UNIX系の場合)
- WMBとは別のOSに配置
 - 複数OSのWMBから同一のデータをアクセスしたい場合に選択(UNIX系の場合)
 - ボトルネックとならないような設計をするように留意

■ 高可用性

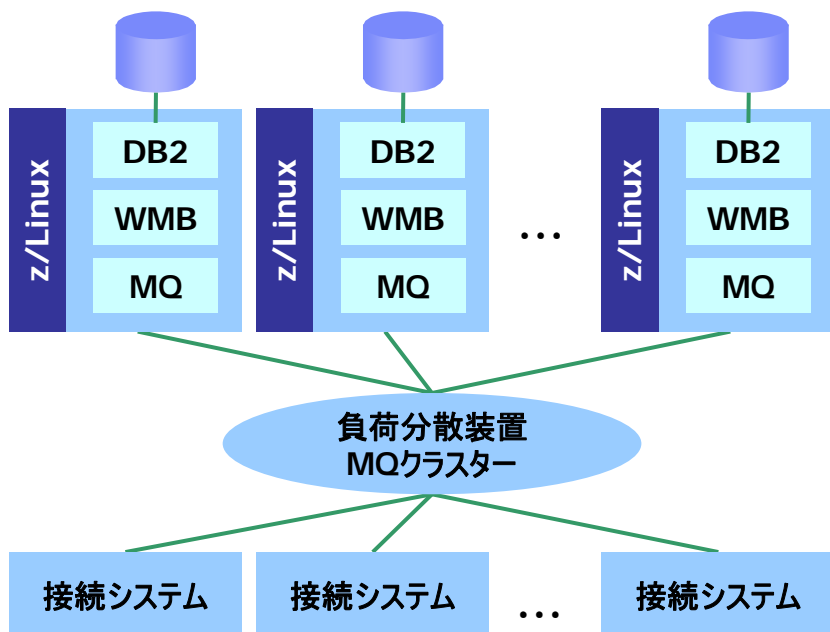
- WMB自身は回復が必要な資源を持たない
 - MQ、データベース管理システムの資源(キュー、DB、ログ)の回復が可能な構成を取る
 - WMBは、Active-Active、Active-Standbyのどちらの構成も可能
 - 障害検知、コンポーネントの再始動は、Linux-HA (Linux)、ARM (z/OS)等で行います

構成例(z/Linux)

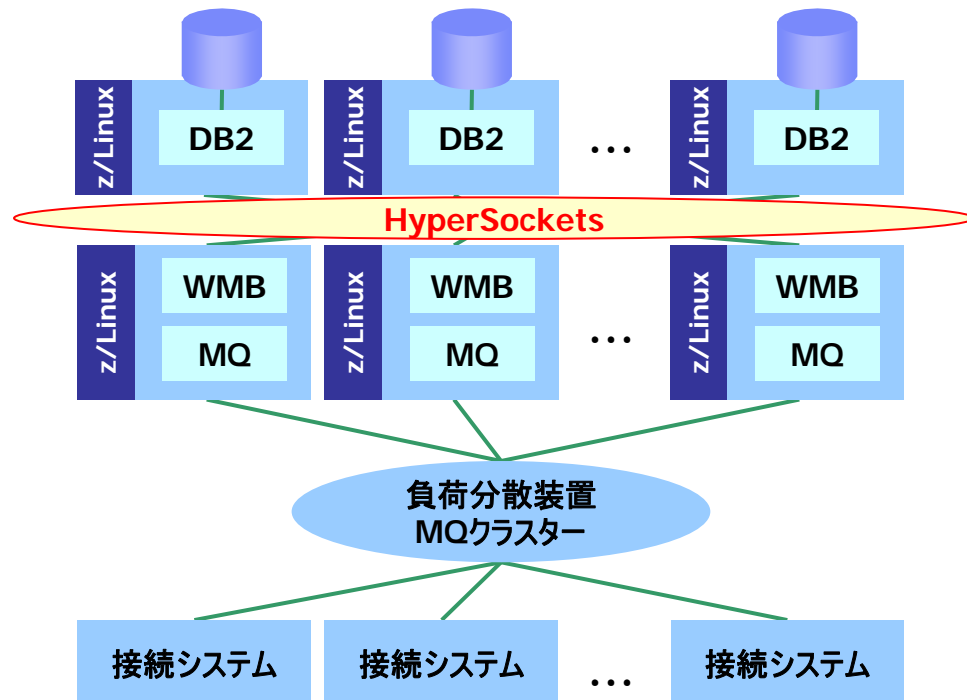
- ③ パフォーマンス
- ④ 高信頼性・高可用性

z/Linuxにおける、高可用性、高スループットを実現するためのWMB構成例を示します。

① ユーザーDBの共有が不要なケース



② ユーザーDBの共有が必要なケース



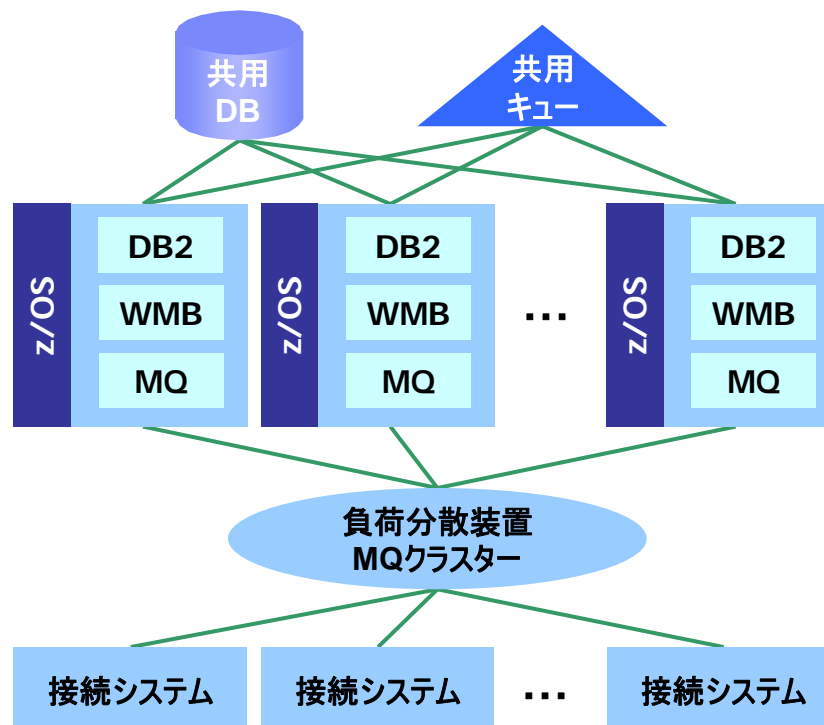
一つのOS内にWMBの実行環境を構成し、冗長構成を取ります。単体ミドルウェアの障害は同一OS内での再始動を行います。OSレベルの障害では、他のActiveなOS、又は、StandbyのOSにてWMB (MQとDB2も)の再始動を行います。

WMB・MQが稼働するOSとDB2が稼働するOSを分離し、それぞれ冗長構成を取ります。同一DBを別OSに配置したWMB間で共有する必要がある場合の構成になります。単体ミドルウェアの障害は同一OS内での再始動を行います。OSレベルの障害では、他のActiveなOS、又は、StandbyのOSにてWMB (MQとDB2も)の再始動を行います。

構成例(z/OS)

- ③ パフォーマンス
- ④ 高信頼性・高可用性

z/OSにおける、高可用性、高スループットを実現するためのWMB構成例を示します。
この構成は他プラットフォームに比べて、最も高い可用性を提供します。



並列Sysplex環境下で、一つのOS内にWMBの実行環境を構成し、冗長構成を取ります。

DBとキューを各OSのWMBから共用可能です。

障害コンポーネントの再始動を待たずして、一部のケースを除き、処理の継続が可能です。

障害コンポーネントの再始動はARMIによって行います。単体コンポーネント障害は同一OSでの再始動、OSレベルの障害では他のOSでWMB (MQとDB2も)を再始動という運用はz/Linuxと同じです

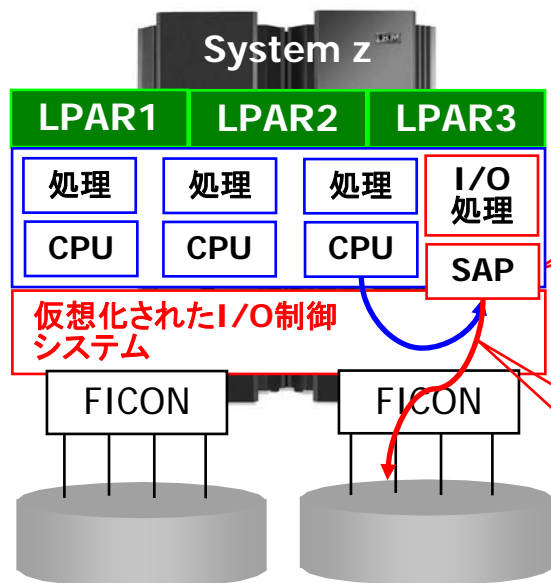
高パフォーマンスを支えるSystem z10の技術

監査ログや取引ログといったDISK I/O性能、他システムとの通信のためのネットワークI/O性能はESBにとって非常に大切です。System zの高性能I/Oによって、CPU資源を無駄なく使う事ができます。

System z10のバランスのとれたシステム設計

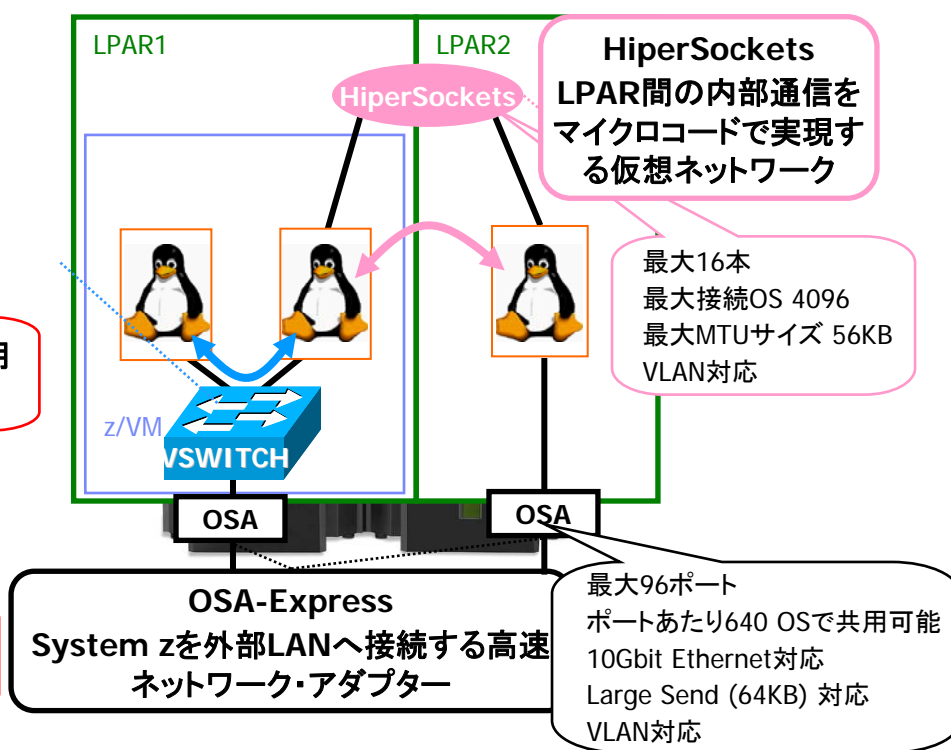
- 高性能プロセッサや大容量メモリの搭載 + ディスクI/O + ネットワークI/O
- プロセッサの使用効率を高め、多数処理の同時実行を可能にする

I/O処理はI/O専用プロセッサに任せ、CPUは業務処理に専念させる事が可能であるためパフォーマンスに影響を及ぼさない



(1) I/O処理専用プロセッサ

(2) I/O処理をH/Wで仮想化



HiperSockets
LPAR間の内部通信をマイクロコードで実現する仮想ネットワーク

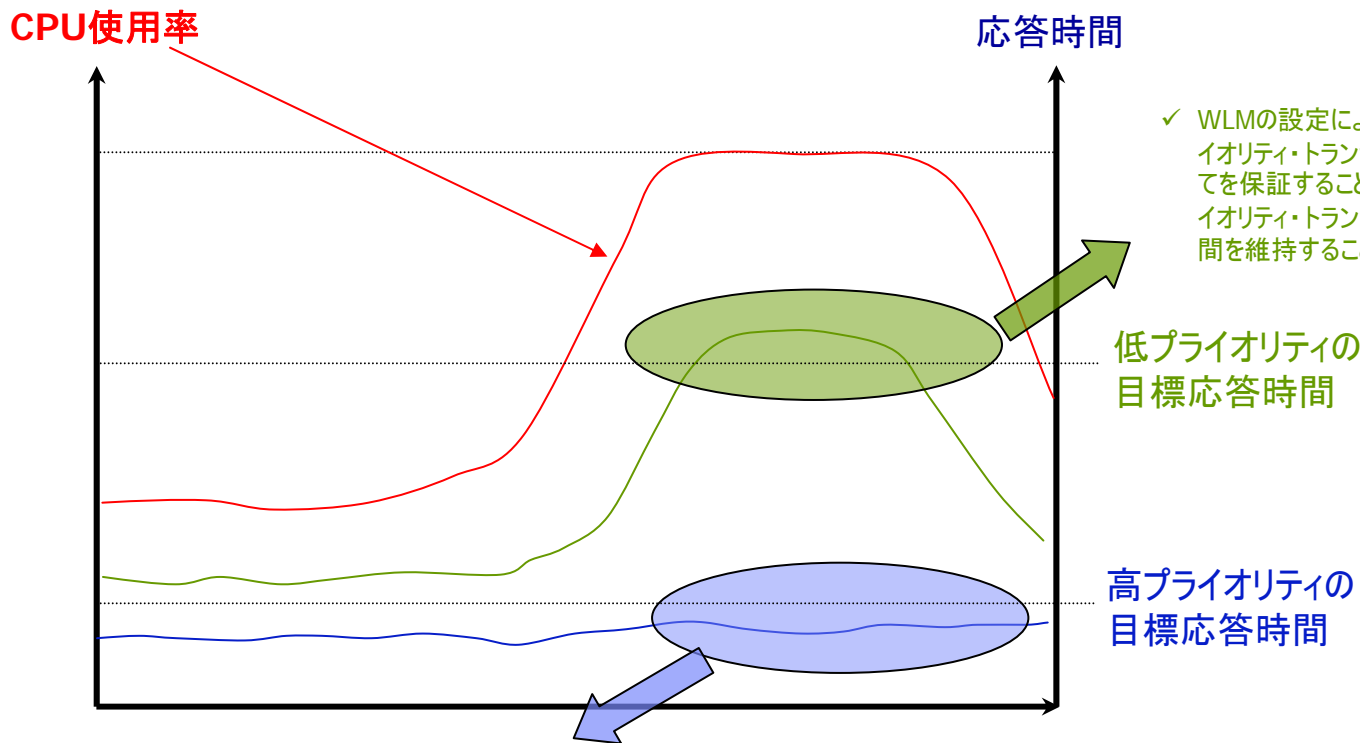
最大16本
最大接続OS 4096
最大MTUサイズ 56KB
VLAN対応

OSA-Express
System zを外部LANへ接続する高速ネットワーク・アダプター

最大96ポート
ポートあたり640 OSで共用可能
10Gbit Ethernet対応
Large Send (64KB) 対応
VLAN対応

z/OS WLMとWMBの連携

WMBは実行グループ単位にWLMポリシーのサービスクラスを割り当てることが可能です。これによって、WMBの処理に優先度を設定する事が可能となります。



✓ WLMの設定により過負荷環境においても、低プライオリティ・トランザクションに最低限の資源割り当てを保証することが可能であり、かつWLMは低プライオリティ・トランザクションに設定された目標応答時間を維持することを試みる

✓ 高負荷状態が高プライオリティ・トランザクションのみではなく、低プライオリティ・トランザクションやバッチ処理で発生している状況であれば、WLMによる適切なワークロード配分や、低プライオリティ・トランザクションの流量制御(最大セッション数)によって、安定した応答時間を目指す設計・運用が可能

費用対効果のマキシマイズ

WMBは下記のSystem zのOn Demand機能を活用して費用対効果のマキシマイズが可能です。

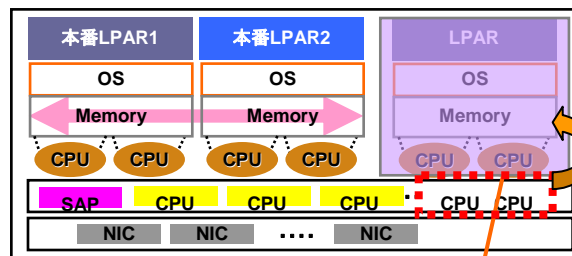
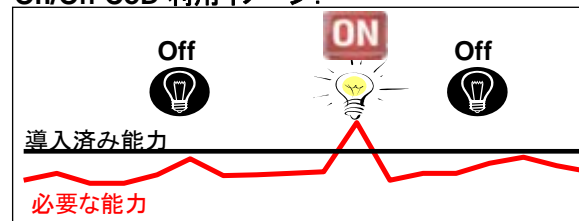
Capacity On Demand

- On/Off Capacity On-Demand
 - 一時的な処理能力増強に対応
- Capacity Backup Upgrade
 - 災害や障害でサーバーの処理能力が失なわれた場合にバックアップ機 (CBU機) のCPU能力を一時的(最長90日間)に増強
- Capacity for Planned Event
 - CBUの適用範囲を拡大し、システム移行やデータセンター移設などの計画停止にバックアップ機のCPU能力を増強
 - 最大3日間の事前設定キャパシティ

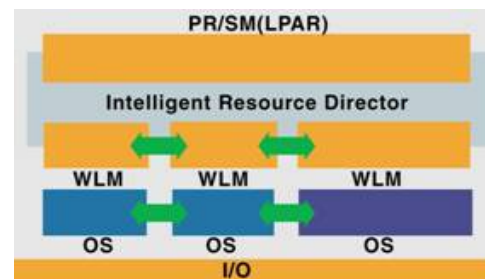
IRD(Intelligent Resource Director)

- CPU、メモリー、I/Oをワークロードに応じて自動的に最適化

On/Off CoD 利用イメージ:

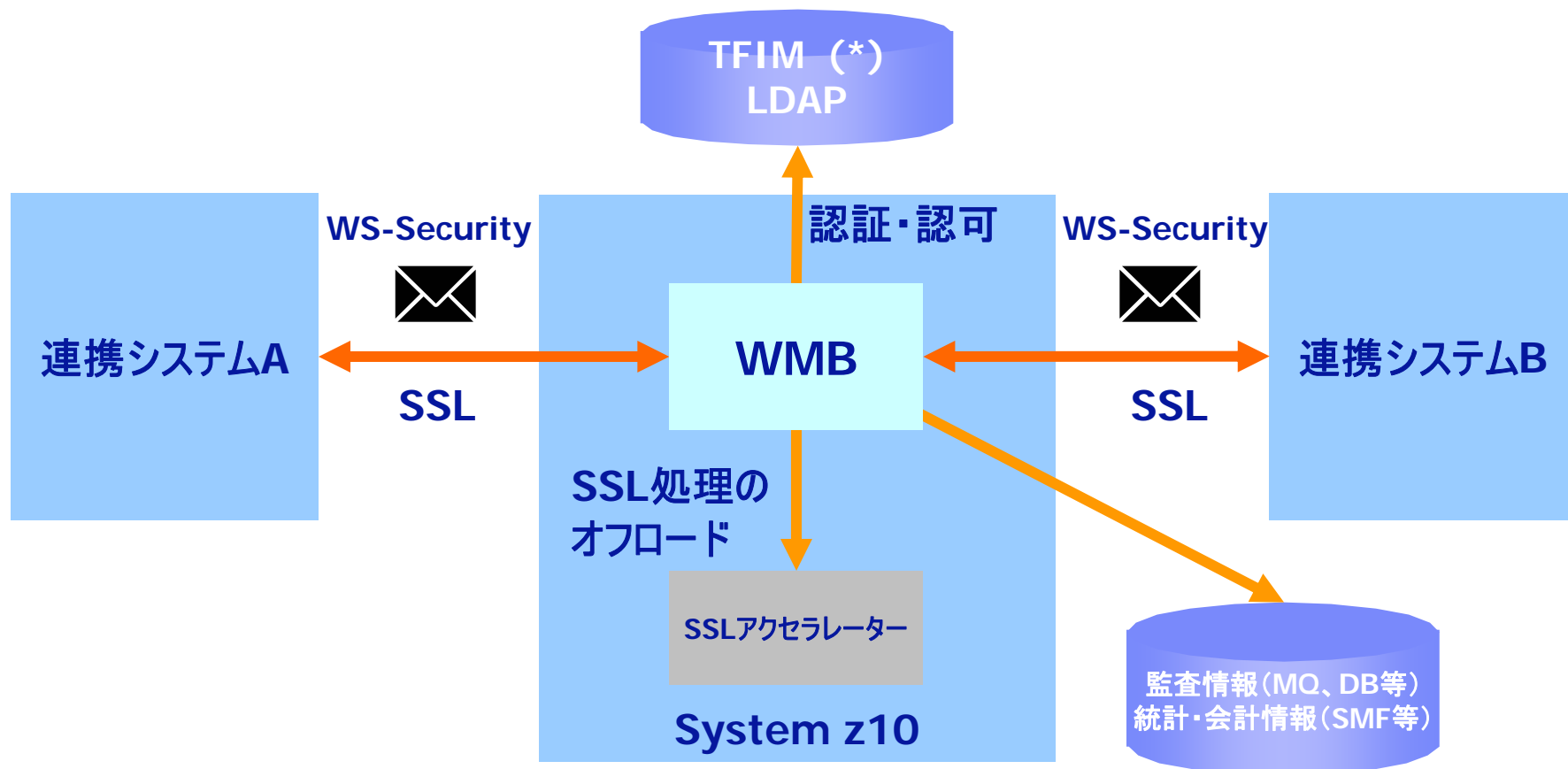


通常は非活性、オンデマンドに活性化



セキュリティ

WMBは仲介処理開始前に、リクエスターの認証・認可を行う事が可能です。また、SSL処理をハードウェアにオフロードする事が可能です。
 また、監視・監査の情報取得、統計・会計情報の取得のためにWMBの標準機能を活用できます。
 統計・会計情報はSMFデータセットへの取得も可能です。



(*) Tivoli Federated Identity Manager

System zとWMBの連携のまとめ

■ z/Linux

- HyperSocketsによる高速通信
- **Capacity on Demandの活用**
- HWセキュリティ機構の活用(主なものは次の2つ)
 - IBM Crypto Express2 (CEX2) (feature 863 for zSeries)
 - SSLのハンドシェイク処理のオフロード
 - IBM CP Assist for Cryptographic Function - CPACF (feature 3863 for zSeries)
 - データ暗号化のアシストを行う(無償フィーチャー)

■ z/OS

- 99.999%の可用性を誇るParallel Sysplexの活用
- HyperSocketsによる高速通信
- **Capacity on Demandの活用**
- **IRDの活用**
- HWセキュリティ機構の活用(z/OSがサポートするものCEX2/CPACFも含む)
- **Java Computeノード、XSLTノード等のzAAPオフロード**
- WLMとの連携
 - 実行グループ単位にパフォーマンス目標を設定可能
- **RRSとの連携**

まとめ

* システム投入のスピードを上げるため、既存資産を有効に活用し、さまざまなシステムを緩やかに連携し、コーディングを最小化するシステム連携基盤の構築がますます重要になってきています。また、パフォーマンス、セキュリティといった非機能要件を満たすためには適切なプラットフォームを選択する事が重要です。

連携基盤に求められること

開発期間の短縮と開発コストの削減

運用・管理コストの削減

処理量増加、接続システム増加に伴う拡張性と可用性

情報漏えい、データ改ざんへの対応と監査情報の取得

ポイント

① 既存資産の最大活用

② 開発生産性
変更容易性

③ 高パフォーマンス
高スループット

④ 信頼性

⑤ セキュリティ

IBMのソリューション

堅牢なプラットフォームで
既存資産を最大活用する
システム連携

ミドルウェアを活用した
脱・手組みのしくみ

System z10
+
**WebSphere
Message
Broker**