



進化し続けるオープンソース アプリケーションサーバ：GlassFish

サン・マイクロシステムズ株式会社
ソフトウェア・ビジネス統括本部
寺田 佳央

<http://blogs.sun.com/yosshi>

Agenda

- はじめに
- Sun SOA 実行基盤のご紹介
- Sun GlassFish ESB のご紹介
- Java EE 6 のご紹介
- GlassFish v3 のご紹介

はじめに

GlassFish: 先進的アプリケーションサーバ



glassfish.dev.java.net

- オープンソースコミュニティ
- アプリケーションサーバ
- Java EE 5 完全準拠
- Web 2.0 対応
 - > Ajax, Comet
 - > Web サービス
- ハイパフォーマンス
- サポート
 - > コミュニティサポート
 - > 商用サポート

GlassFish を選択する理由

OSS だから

ソースコードの閲覧が可能
発表前の製品を評価可能
活発なプロジェクト活動

TCO の削減

開発時は無料で利用可能
目的に応じたライセンス形態
運用もかんたん（教育コスト）

オープンソース
+
Sun のバリュー

高性能・高信頼

数多くの大規模導入事例
他の商用製品を凌ぐベンチマーク結果
実績ある商用製品がベース

Sun は OSS コミュニティに最も貢献している一社
有償ユーザには Sun が訴訟のリスクをカバー

GlassFish を選択した理由（お客様の声）

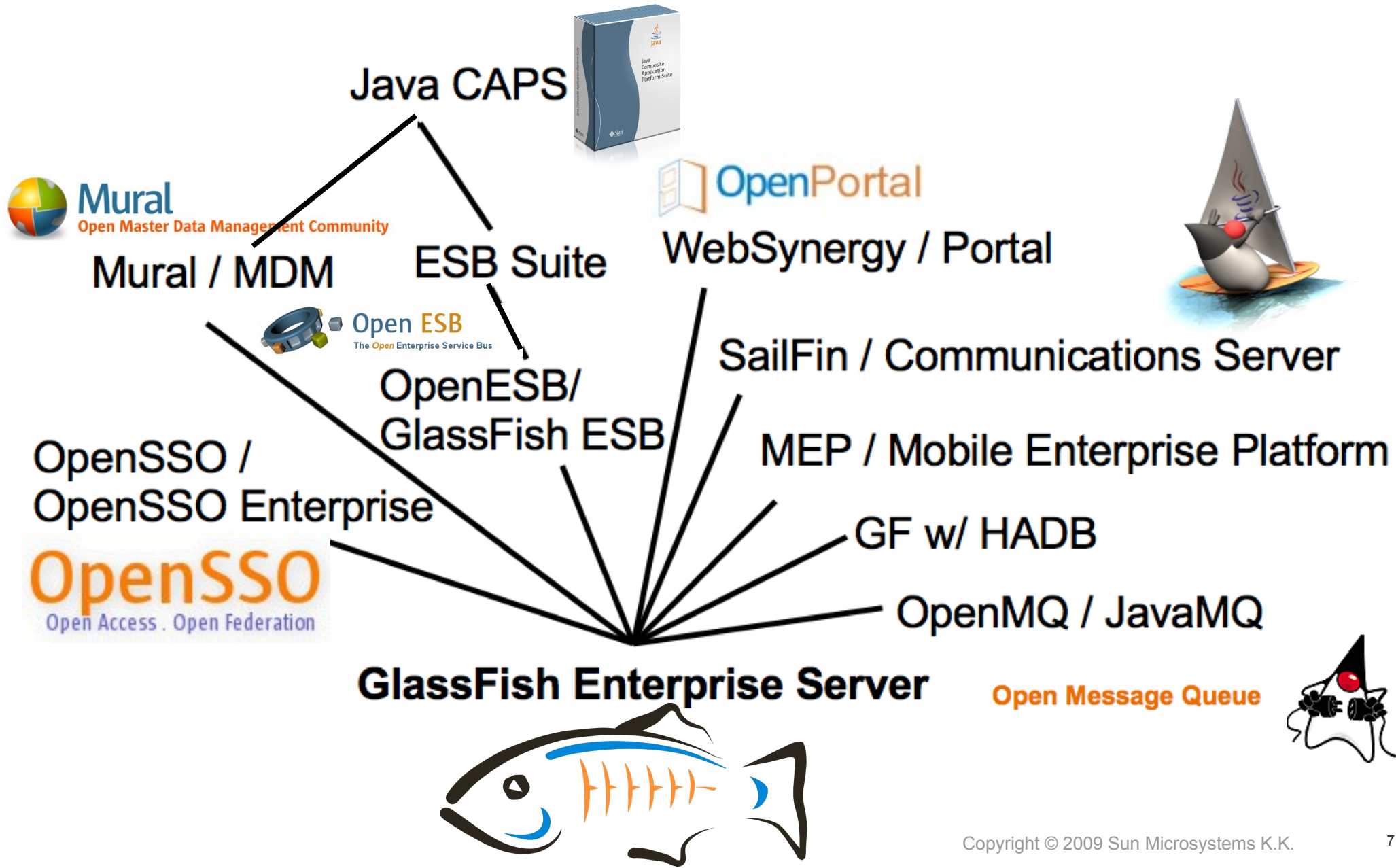
GlassFishのどこが一番気に入っていますか？

- > Tomcat はできないことがある。
- > Weblogic はときどき大げさになってしまったりする。
- > GlassFish はこれからでしょうが、ほどよい感じです。

参考 URL :

http://blogs.sun.com/stories/resource/rakuten/Rakuten-glassfish-questionnaire_ja.html

製品の中で重要な役割を果たす GlassFish



コミュニティと Sun の付加価値

無償ユーザ



- ・ オープン開発
- ・ 公開されたテスト
- ・ フィードバック
- ・ 無料で利用可能
- ・ メーリングリスト
- ・ フォーラムサポート

有償ユーザ



- ・ 評価済みの製品
- ・ サブスクリプション
- ・ パーペチュアル
- ・ 運用サポート
- ・ バグ修正
- ・ パッチ提供

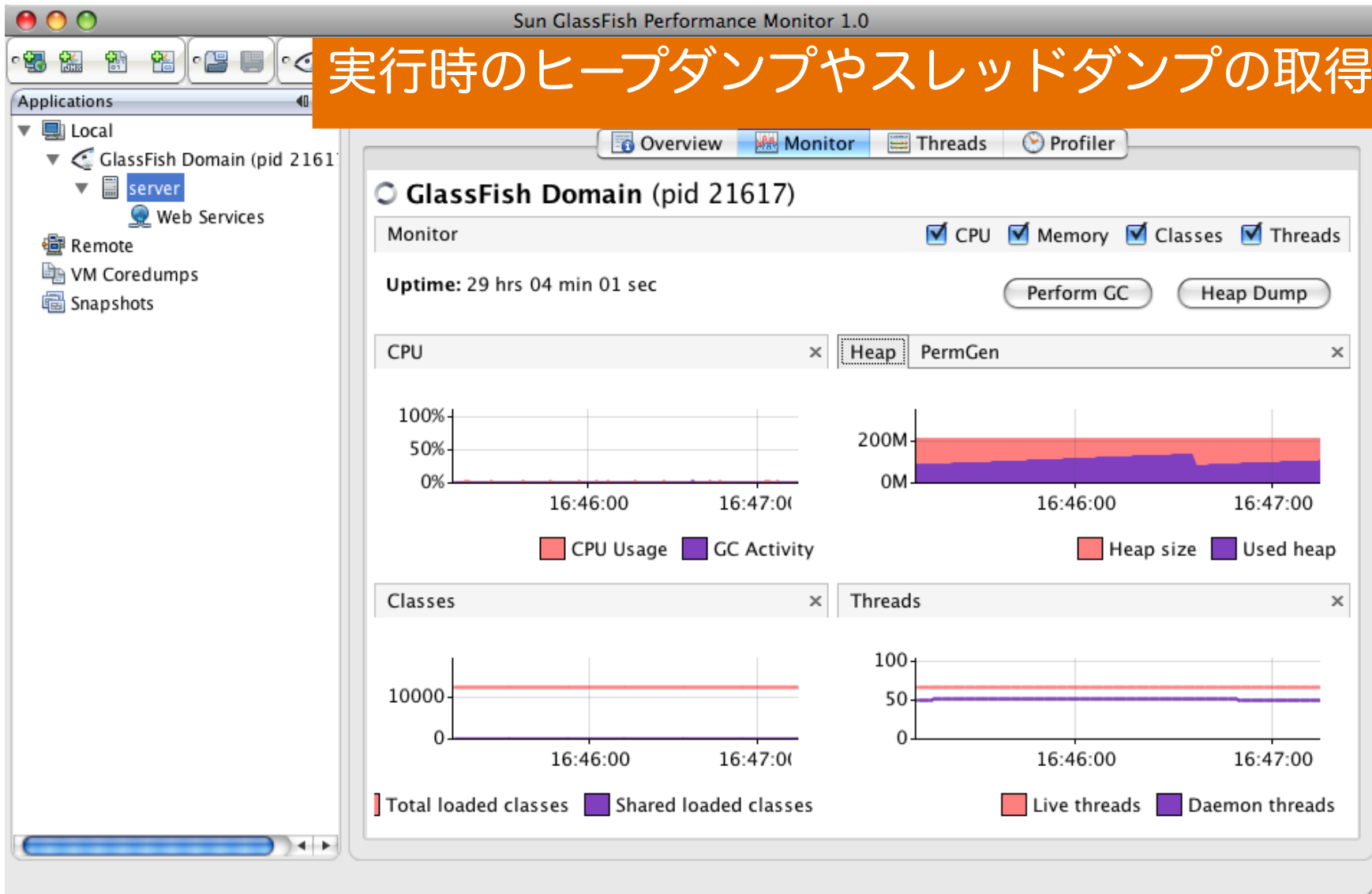
プラス α の付加価値



- ・ トレーニング
- ・ コンサルティング
- ・ OEM ライセンス契約
- ・ 特別サポート契約

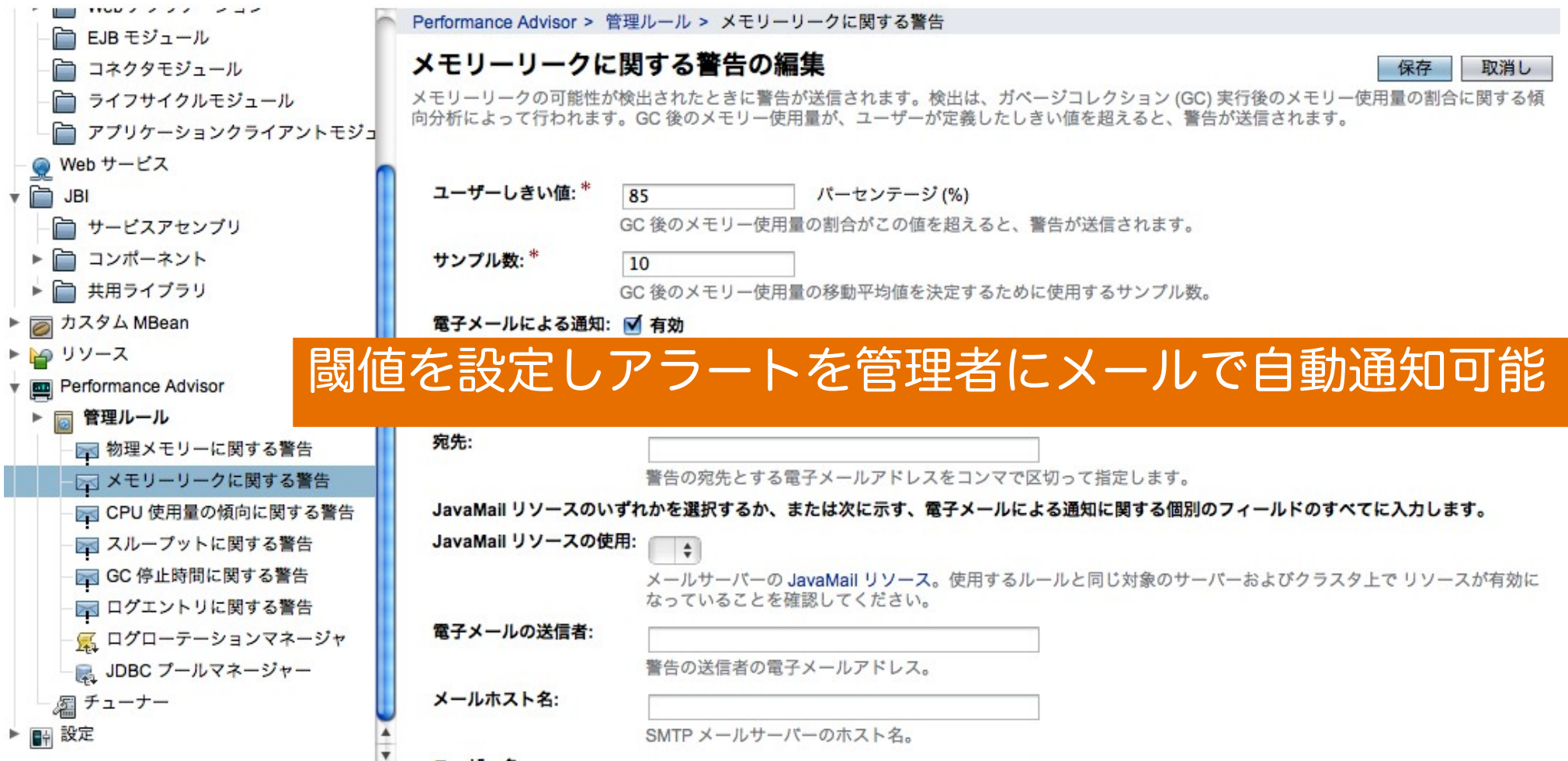
有償ユーザに提供される付加価値 パフォーマンス監視ツール

実行時のヒープダンプやスレッドダンプの取得も可能



有償ユーザに提供される付加価値

パフォーマンスアドバイザー



メモリーリークに関する警告の編集 [保存] [取消し]

メモリーリークの可能性が検出されたときに警告が送信されます。検出は、ガベージコレクション (GC) 実行後のメモリー使用量の割合に関する傾向分析によって行われます。GC 後のメモリー使用量が、ユーザーが定義したしきい値を超えると、警告が送信されます。

ユーザーしきい値: * パーセンテージ (%)

GC 後のメモリー使用量の割合がこの値を超えると、警告が送信されます。

サンプル数: *

GC 後のメモリー使用量の移動平均値を決定するために使用するサンプル数。

電子メールによる通知: ☒ 有効

閾値を設定しアラートを管理者にメールで自動通知可能

宛先:

警告の宛先とする電子メールアドレスをコンマで区切って指定します。

JavaMail リソースのいずれかを選択するか、または次に示す、電子メールによる通知に関する個別のフィールドのすべてに入力します。

JavaMail リソースの使用:

メールサーバーの JavaMail リソース。使用するルールと同じ対象のサーバーおよびクラスタ上で リソースが有効になっていることを確認してください。

電子メールの送信者:

警告の送信者の電子メールアドレス。

メールホスト名:

SMTP メールサーバーのホスト名。

有償ユーザに提供される付加価値

SNMP 監視対応

例 1 : サーバの状態の監視

```
# snmpwalk -c public -v 1 -m ./J2EE-MIB localhost:10161
J2EE-MIB::j2eeSrvSMState
J2EE-MIB::j2eeSrvSMState.1.1 = INTEGER: running(4)
```

例 2 : デプロイされているアプリケーション数

```
# snmpwalk -c public -v 1 -m ./J2EE-MIB localhost:10161
J2EE-MIB::j2eeAppMoName.1
J2EE-MIB::j2eeAppMoName.1.2 = STRING:
"name=__JWSappclients,server=server"
J2EE-MIB::j2eeAppMoName.1.3 = STRING:
"name=MEjbApp,server=server"
```

例 3 : アプリケーションが起動した時間

```
# snmpwalk -c public -v 1 -m ./J2EE-MIB localhost:10161
J2EE-MIB::j2eeAppSMStartTime.1
J2EE-MIB::j2eeAppSMStartTime.1.2 = STRING: "Fri Feb 13
16:43:11 JST 2009"
```

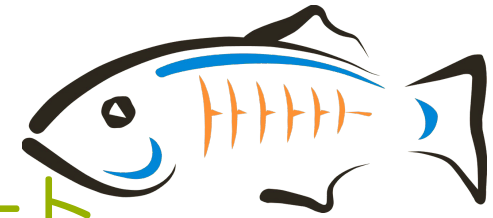
GlassFish の導入事例

<http://blogs.sun.com/stories>

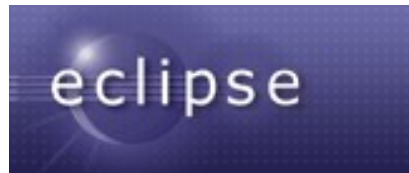


開発環境と実行環境

さまざまな統合開発環境と実行環境をサポート



NetBeans



GlassFish ポートフォリオ



Sun GlassFish Portfolio

GlassFish
Enterprise
Server

GlassFish
Web
Stack

GlassFish
ESB

GlassFish
Web Space
Server

複数の製品を一つのライセンスで提供

- 投資費用 対 効果

- ＞ スケーラビリティ

- ＞ 圧倒的な低価格 / コストパフォーマンス

GlassFish ポートフォリオ

Web関連の製品を組み合わせた製品ラインアップ

GlassFish
Enterprise
Server

高機能、高信頼性、ハイパフォーマンスを兼ね備えた
先進的アプリケーションサーバ

GlassFish
Web
Stack

LAMP/SAMP スタック
Apache , PHP, Ruby, MySQL Community, Memcached,
Tomcat, Lighttpd, Squid, and more.

GlassFish
ESB

軽量な SOA の実行基盤
JBI 準拠の Open ESB を GlassFish に統合

GlassFish
Web Space
Server

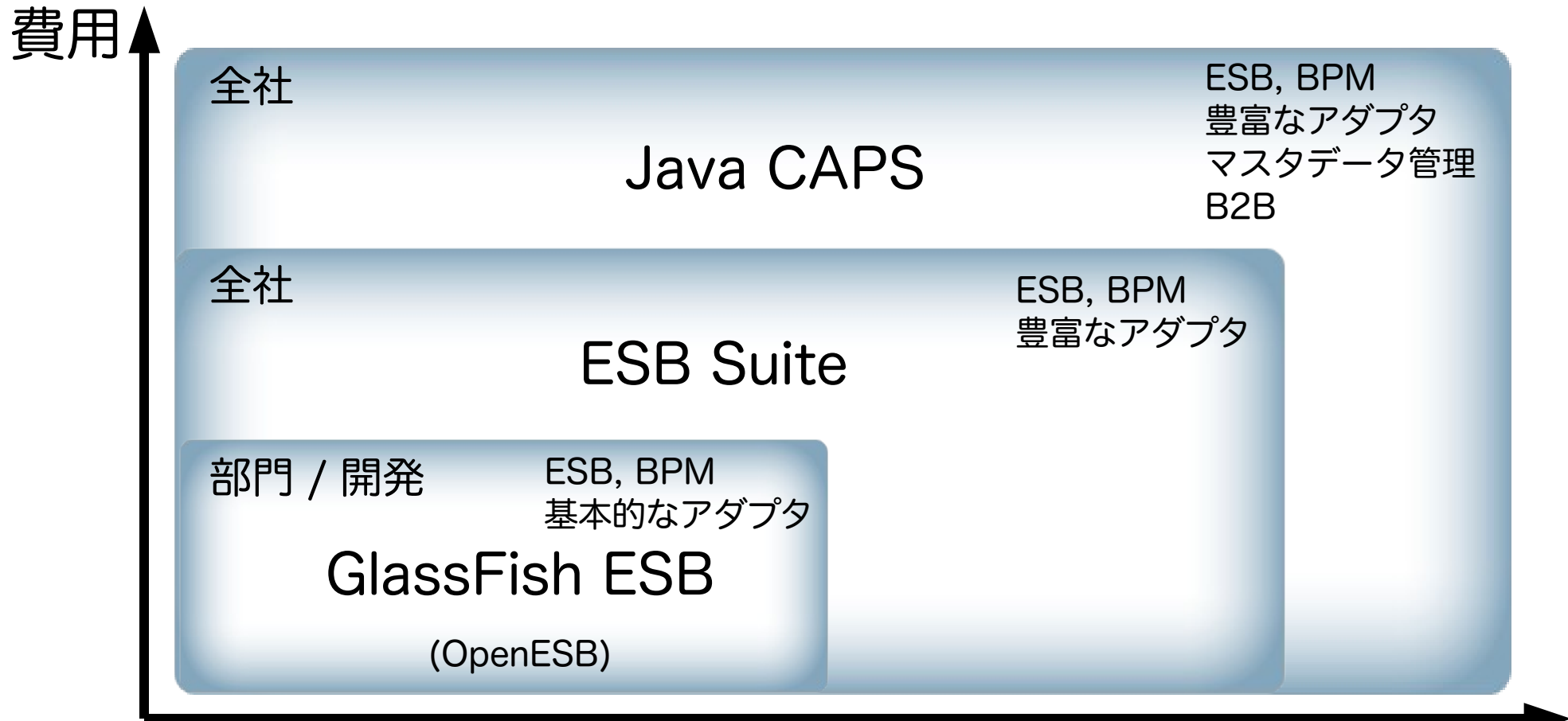
ポータルサーバ
オープンソースポータル Liferay をベース

Sun SOA

実行基盤のご紹介

全体最適を実現する Sun の SOA 製品群

業務要件に応じた選択が可能



- スモールスタートで初期投資削減, リスク軽減
- 適材適所で TCO 削減

OpenESB と GlassFish ESB の違いについて

- OpenESB

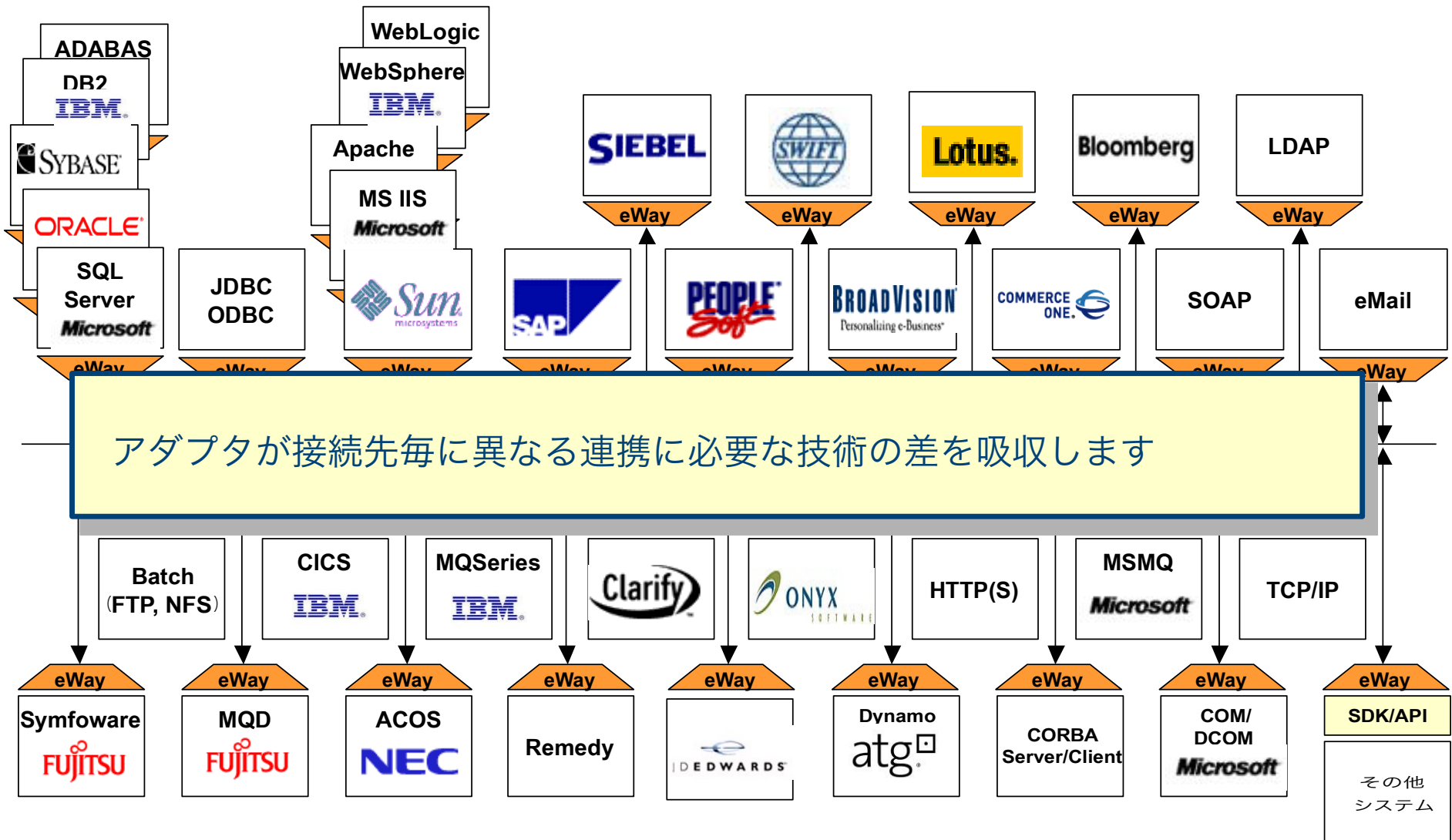
- > Sun のオープンソースの JBI(JSR-208) 実装
- > 商用サポート無し
- > GlassFish 以外のアプリケーションサーバでも動作可能

- GlassFish ESB

- > GlassFish + OpenESB
- > Sun からの商用サポートがあり
- > GlassFish の管理画面で OpenESB 用のカスタマイズあり
- > 汎用機や ERP 等との接続、もしくはプレミアムサポートが必要ない場合

GlassFish ESB と Sun Java CAPS の違い

Sun Java CAPS が提供する豊富な連携用アダプタ



* 上記のアダプタには日本語版未リリースも含まれます

* Sun Java CAPS ESB Suite ライセンスに含まれないアダプタ (別途有償) も記載しております

Sun GlassFish ESB のご紹介

GlassFish ESB の特徴



GlassFish ESB

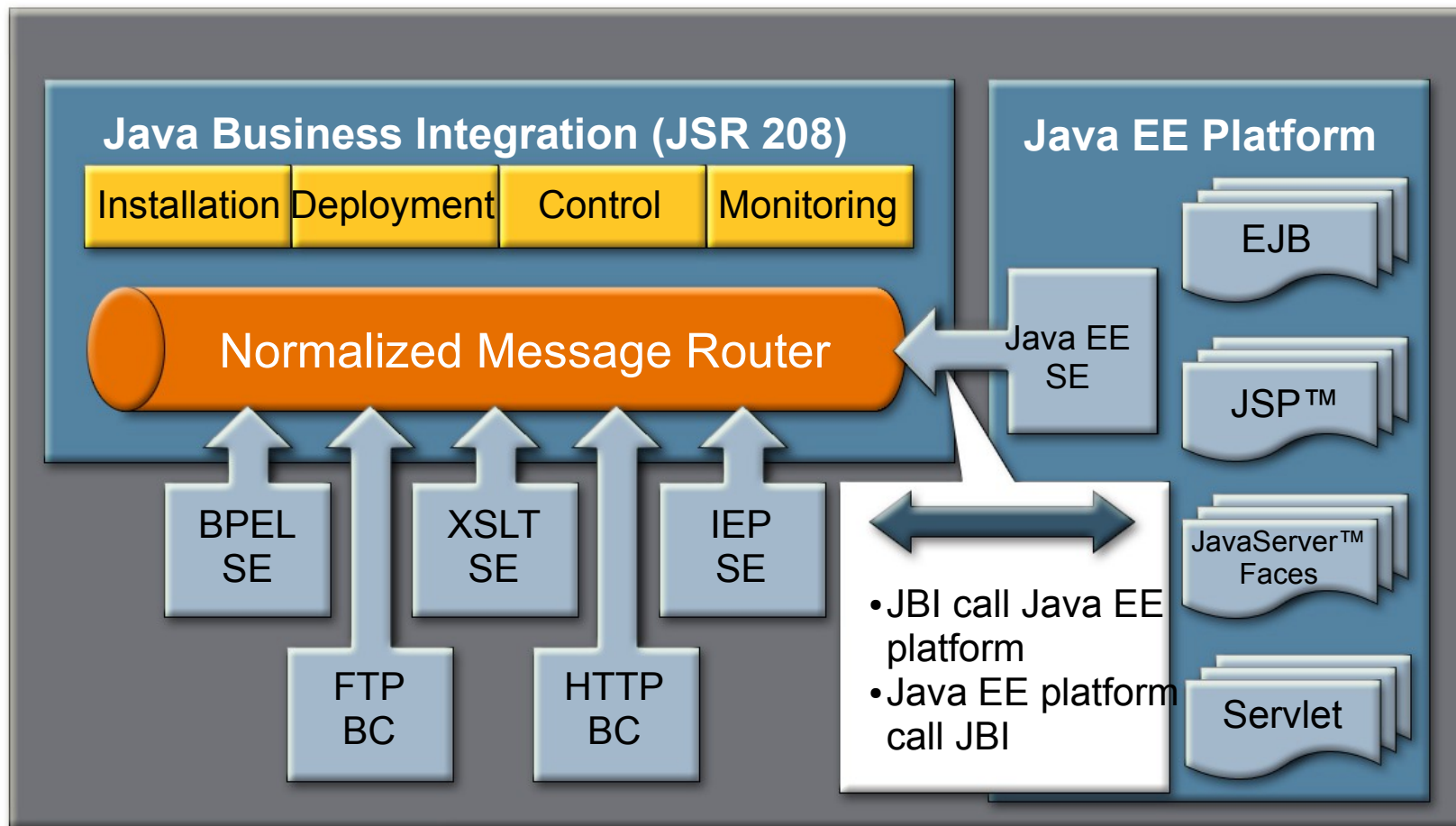
- 低価格
 - > 低価格のサブスクリプションによるライセンス形態
- オープンソース／オープンコミュニティ
 - > Open ESB コミュニティの成果物を利用
- エンタープライズ環境をサポート
 - > Sun Microsystems によるあんしんサポート
- 標準仕様に基づくプラグイン可能な製品
 - > Java Business Integration – JBI (JSR-208)
- かんたん、柔軟な開発環境
 - > Java EE と GUI ツールをもつかんたん開発環境

JBI(Java Business Integration) を統合

JBI とは (JSR-208)

- ベンダーロックインを排除したビジネス統合を行うための標準仕様
- SOAP に基づく Web サービス仕様を現実のビジネス統合できるように改善
 - XML で Web サービスを定義する事でプラットフォームや開発言語に依存しないシステム連携が可能になった
 - 現実のビジネスでは HTTP の同期処理だけではない
 - Web サービスにおける非同期処理を実現

JB1 アーキテクチャ

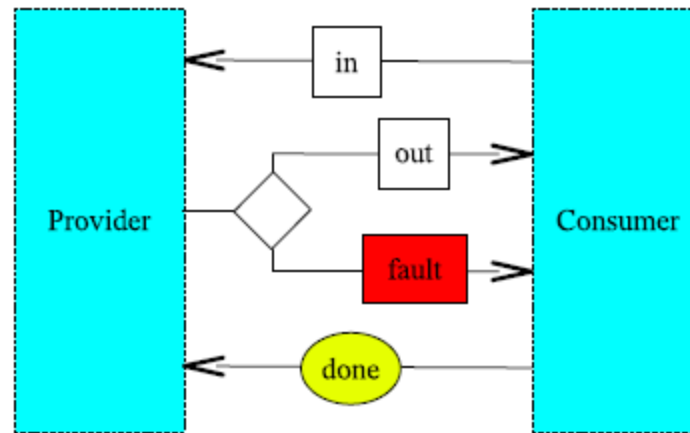


- SE : JB1 内部向けにビジネスロジックや変換サービスを提供 (Java)
- BC : 外部サービスへの接続 (HTTP,FTP,DB,LDAP,JMS) (非 java)
- NMR : SE,BC 間の全てのメッセージは NMR を経由し粗結合を実現

メッセージ交換パターン

メッセージ交換で粗結合を実現

- WSDL でメッセージ交換の操作を定義
- NMR を通じてコンポーネント間でメッセージを配信
 - > 一方はサービス利用者
 - > 一方はサービス提供者



in-out message exchange pattern

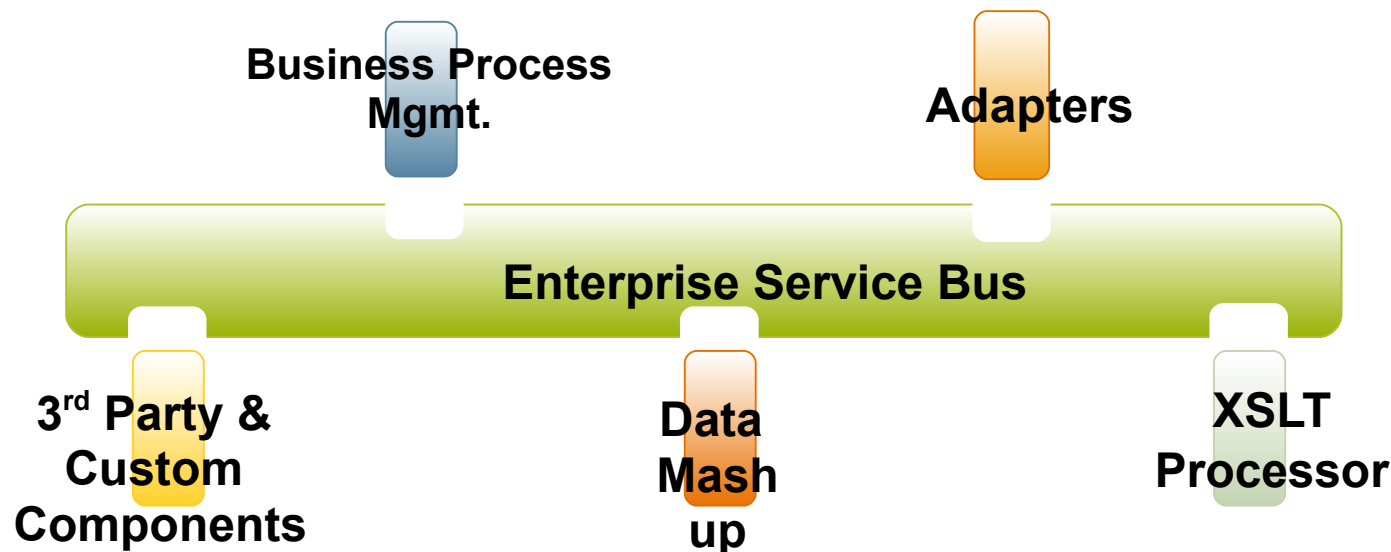
- NMR 経由で交換されるメッセージは「標準化されたフォーマット」を使用
- JBI 1.0 のメッセージ交換は永続化されていない

ESB(Enterprise Service Bus)

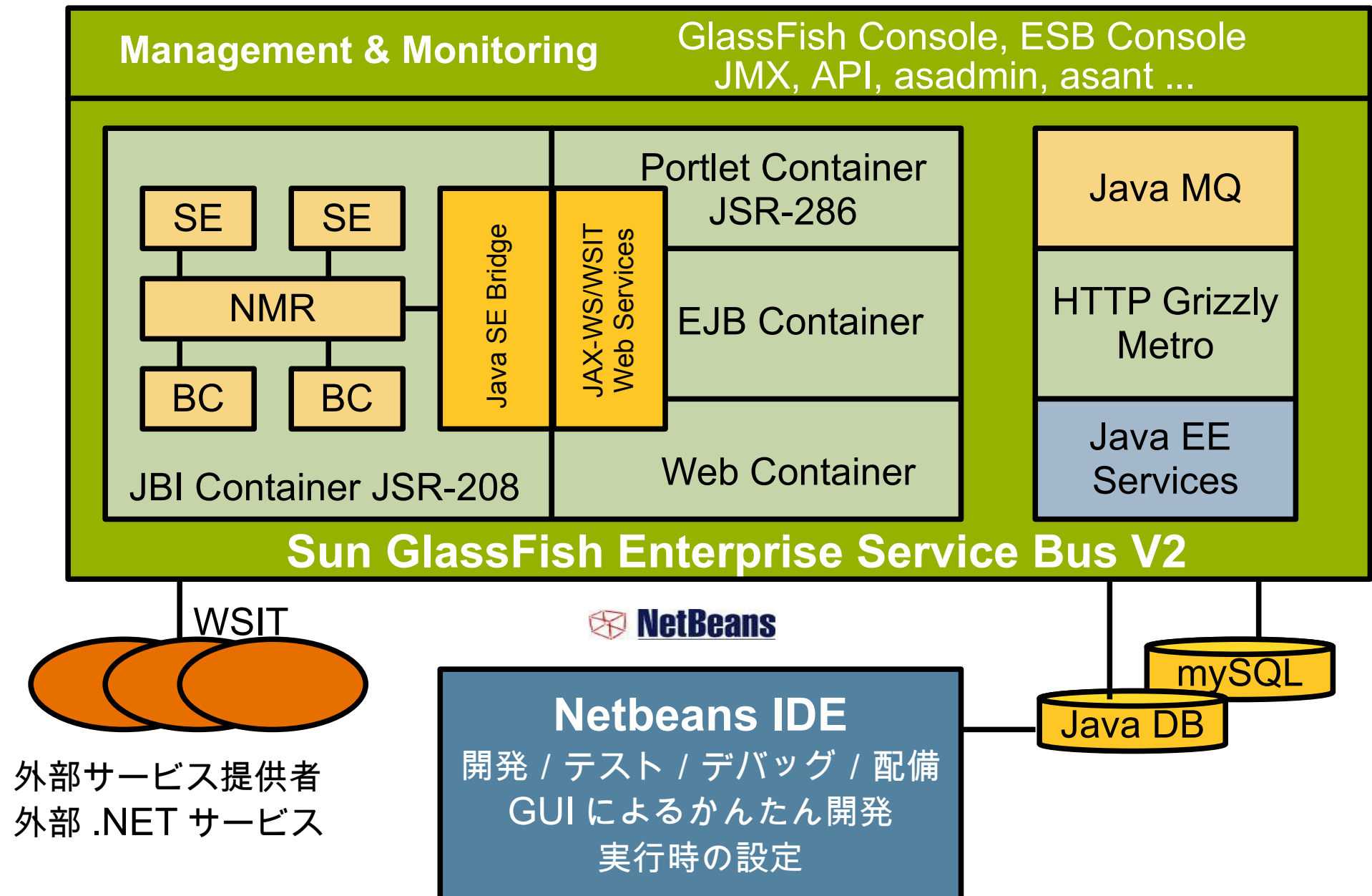
“An ESB is a **Web-services**-capable **middleware** infrastructure that supports intelligent **program-to-program communication** and mediates the relationships among **loosely-coupled** (SOA) and **uncoupled** (event-driven) business components.”

- Roy Schulte

Gartner

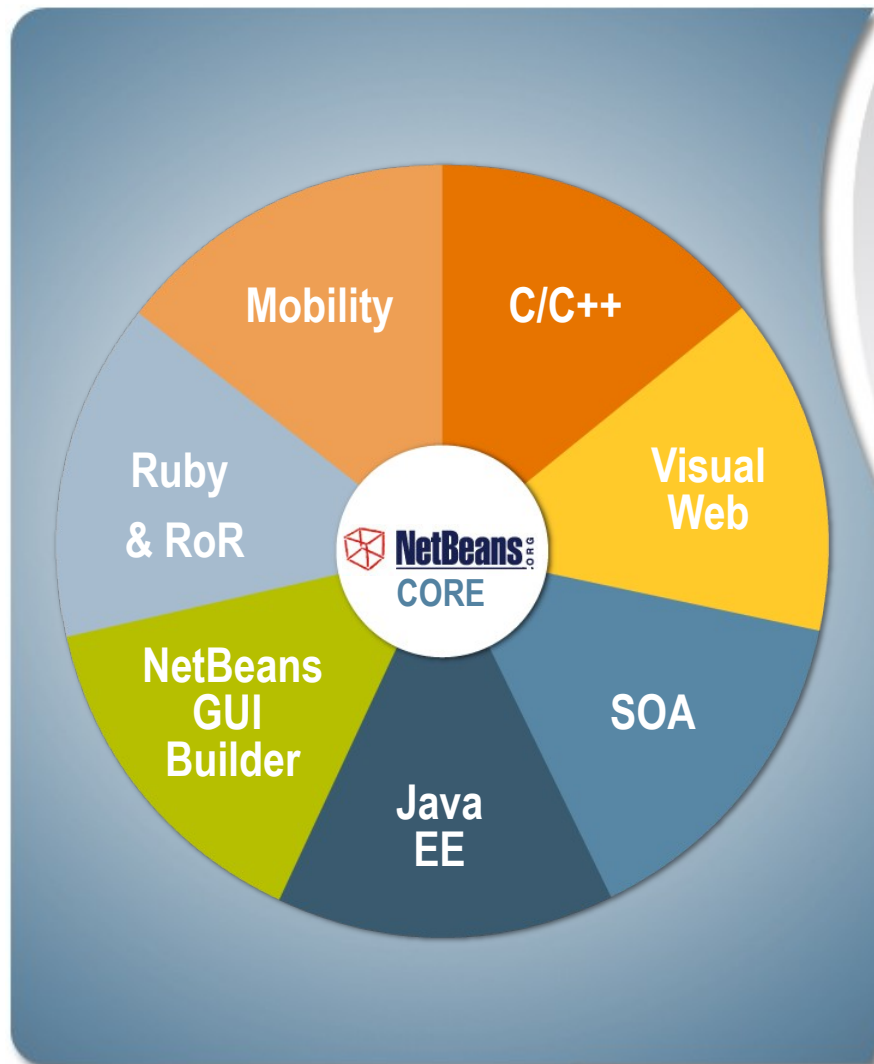


GlassFish ESB コンポーネント



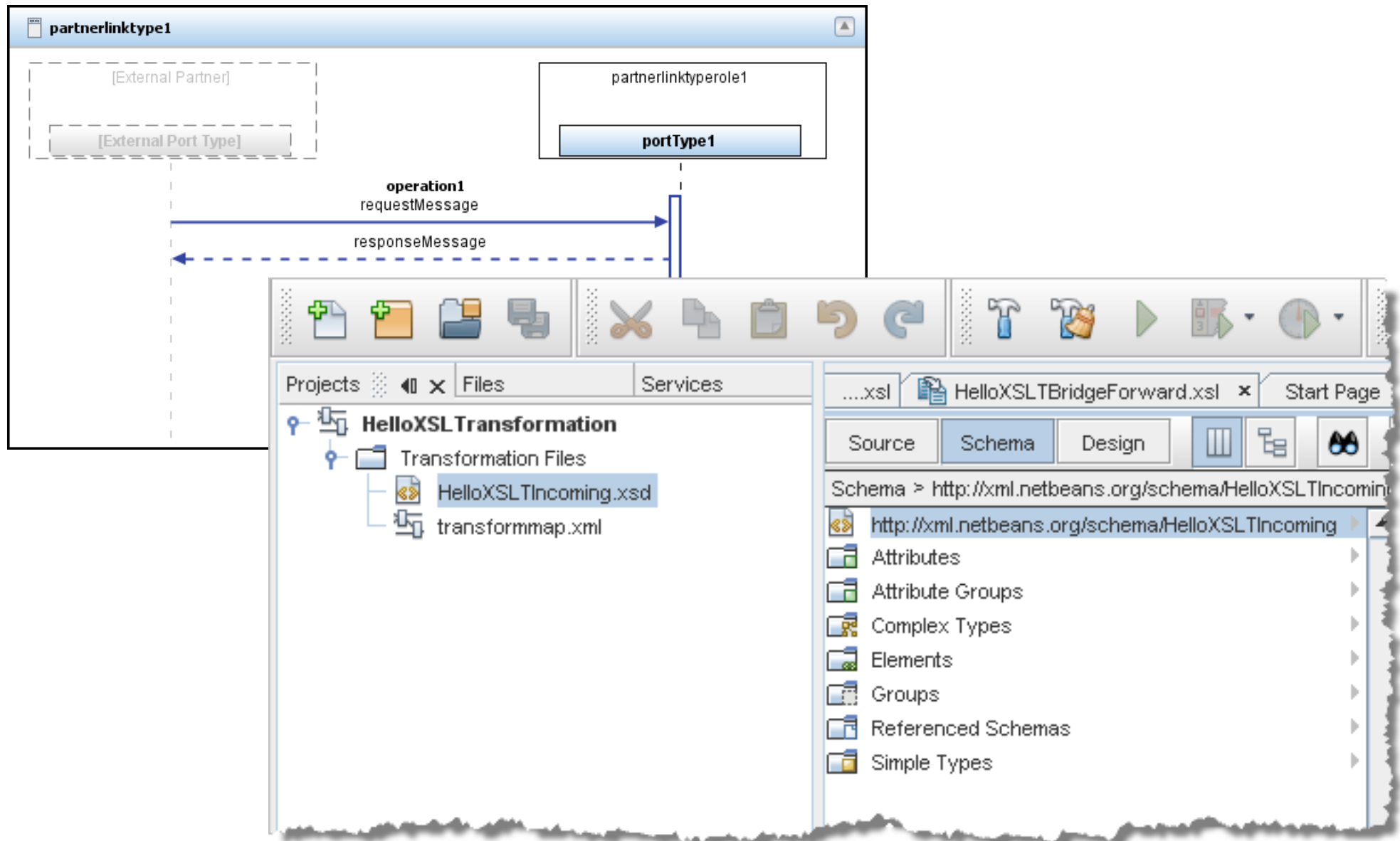
開発 / デバッグ / 配備

NetBeans 統合開発環境



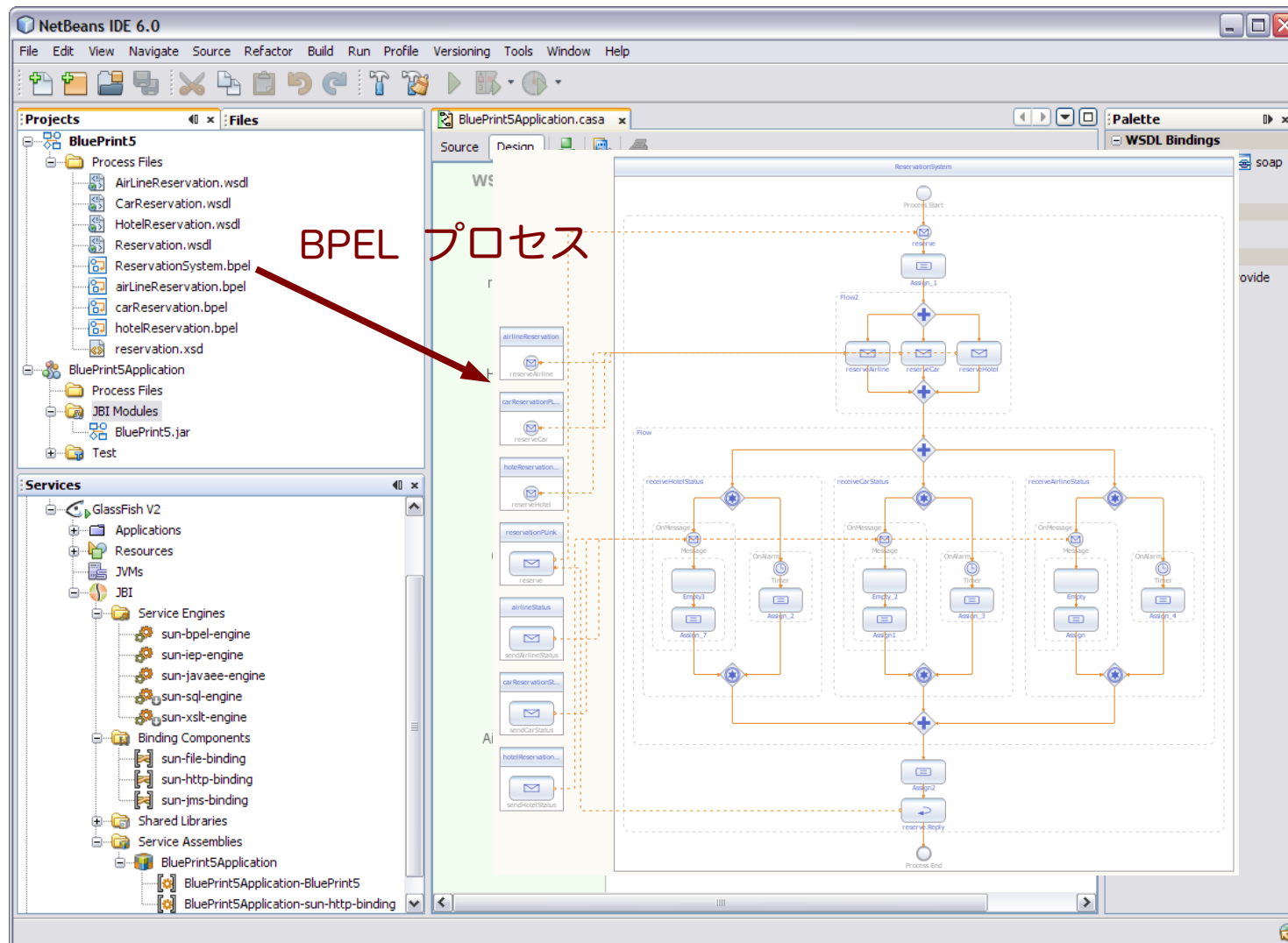
- 簡単インストール
- 100% Pure Java
- Swing GUI ビルダー
- Mobility
- Java EE 5
 - > GlassFish V2 バンドル
- Ruby/Ruby on Rails
 - > JRuby1.1 バンドル

WSDL エディタ /XML スキーマ作成



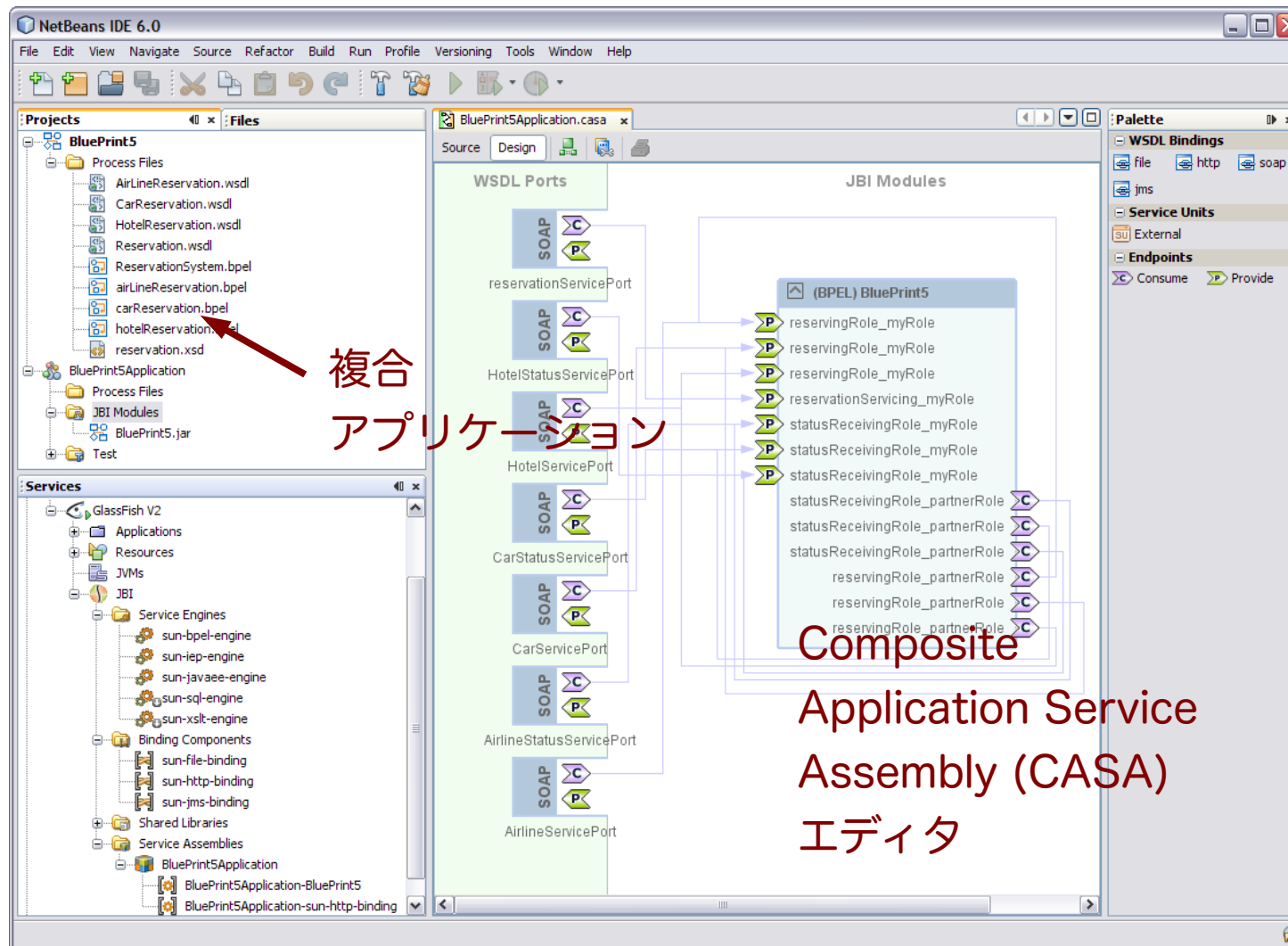
BPEL 2.0 エディタ

グラフィカルにビジネスプロセスのワークフローを記述



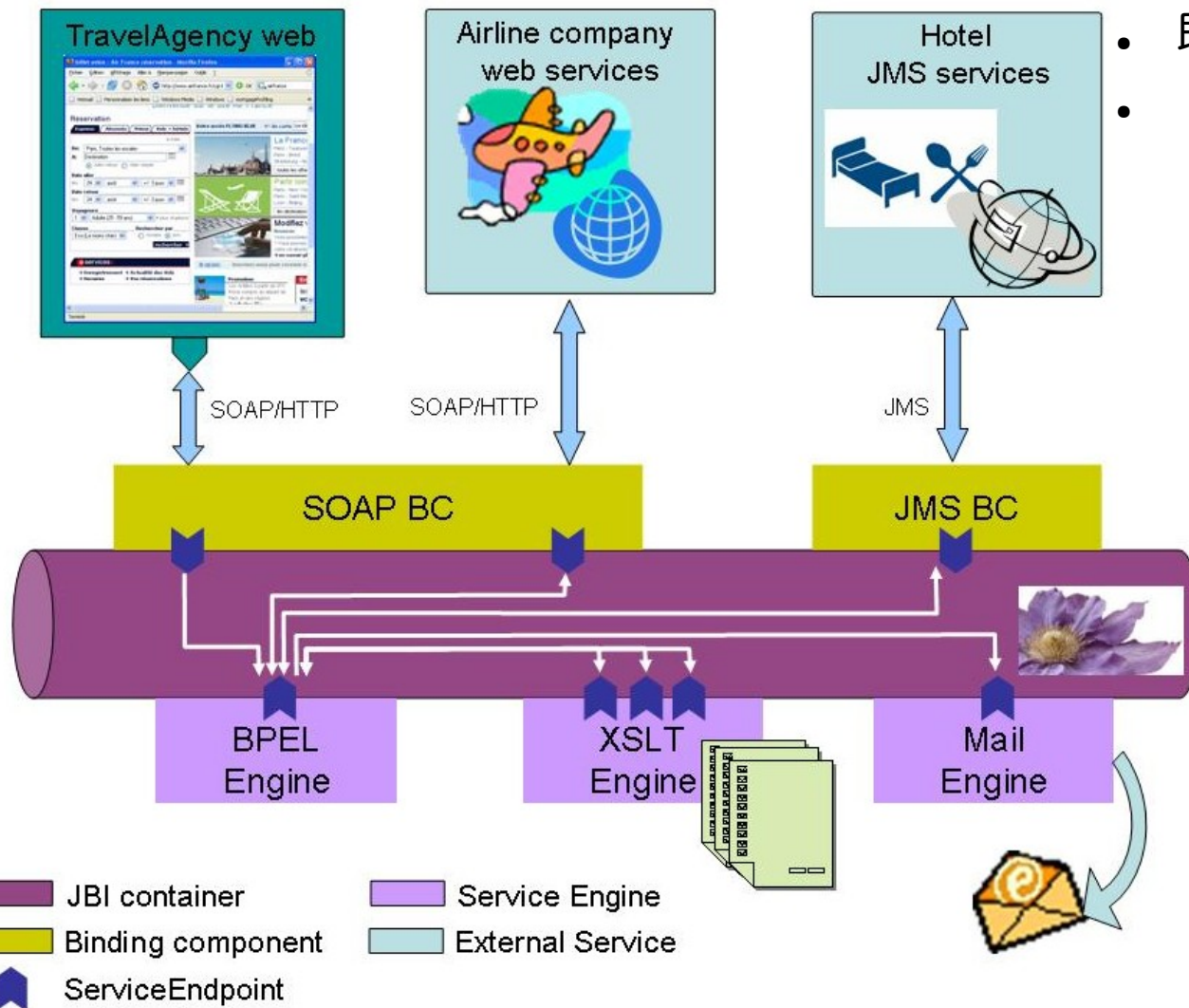
CASA エディタ

複合アプリケーションの関連/構成を記述



GlassFish ESB で適用するシステム

企業間サービスの連携



- 既存の Java EE との連携
- HTTP, MQ, DB, Mail 等 GlassFish ESB が提供するコンポーネントで接続可能な場合

GlassFish ESB が提供するコンポーネント

- バインディング (結合) コンポーネント
 - HTTP
 - File
 - JMS
 - FTP
 - Database
 - LDAP
- サービスエンジン
 - Java EE
 - BPEL
 - XSLT
 - Data Mashup
- ウィザード & Add-ons
 - JMS JCA
 - XSLT Editor
 - Custom Encoder
 - Binding Wizard
 - JAXB Code Seeder

Sun Java CAPS で適用するシステム

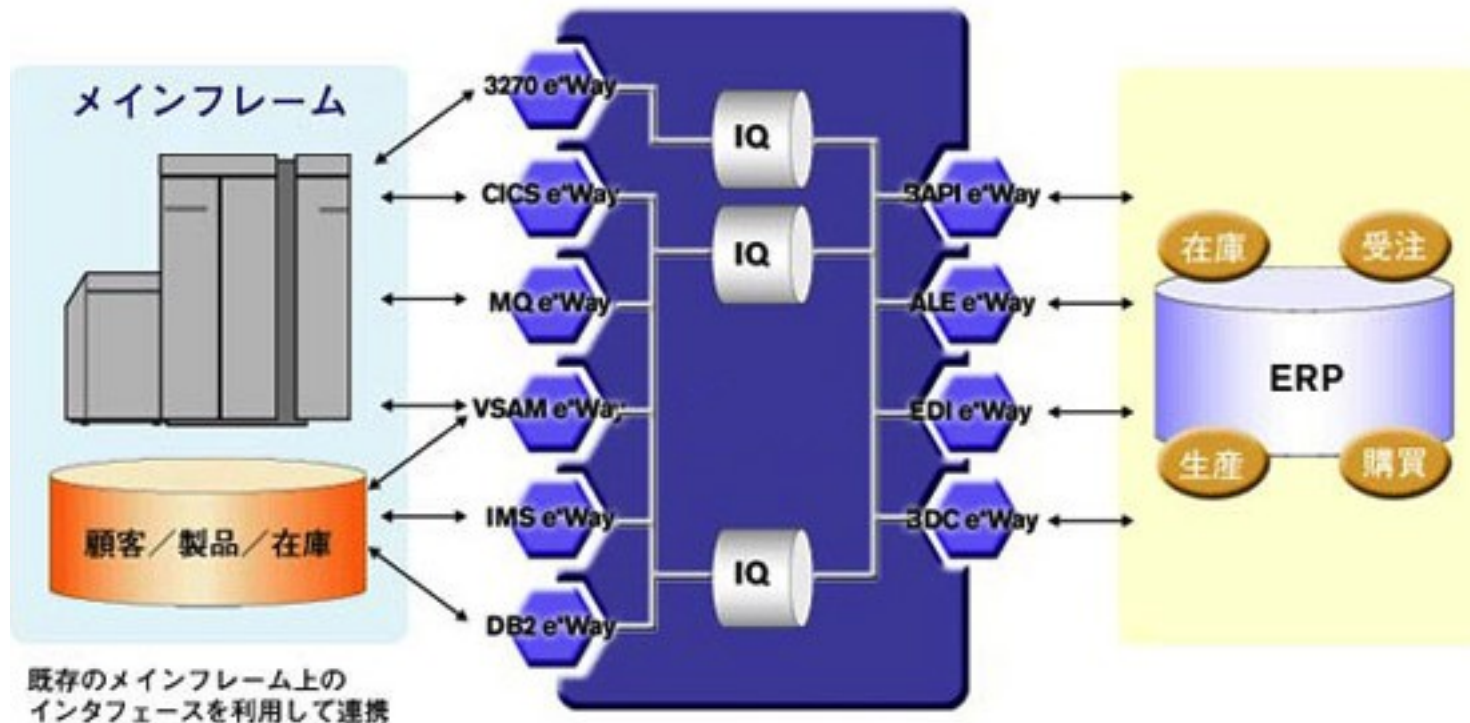
新規 ERP 導入によるシステム連携



- CAPS を使用し既存システムからのデータ移行が容易
- データフォーマットや処理タイミングの差を吸収
- 既存システムからの早期な移行が可能

Sun Java CAPS で適用するシステム

メインフレーム、リアルタイム連携



- システムの構成変更を行わず相互運用が可能
- マスターデータの同時更新、同期、逐次処理が可能

Sun GlassFish ESB のまとめ

利点

- 小規模に低価格から SOA を開始可能
- SOA の実現規模 / ニーズに応じた製品選択が可能
- オープンソースをベース
- ハイパフォーマンス
- ハイクオリティ
- 高信頼性

Sun Java CAPS を利用すると大規模 SOA の実現も可能

- 金融系取引で必要となる SWIFT アライアンス対応
- 汎用機 /SAP 等のシステムとの連携も可能

Java EE 6 のご紹介

Java EE 6 のテーマ

Right Sizing

- 拡張性
- プロファイル
- 仕様の削減
- 進化するかんたん開発

拡張性

> Java EE の肥大の抑制

> Java Ee が全ての技術的な要望をカバーし拡大する時代は終わり

> 既存技術の有効活用

> オープンソースのライブラリやフレームワークの取り込み

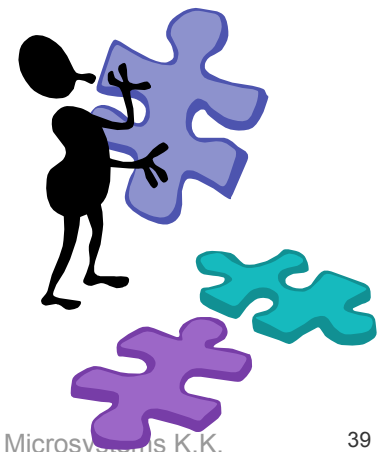
> プラグインとして有効活用



プロファイル

Java EE のサブセットを提供

- Java EE の技術を用途に応じ一部を選択して利用
- ベンダーが Java EE の機能を柔軟に提供可能
- プラットフォームに応じてルール設定が可能
- 独自にプロファイルの開発が可能
 - > 電話会社向けプロファイル
- Java EE 6 で最初に提供されるプロファイル
 - > Web プロファイル



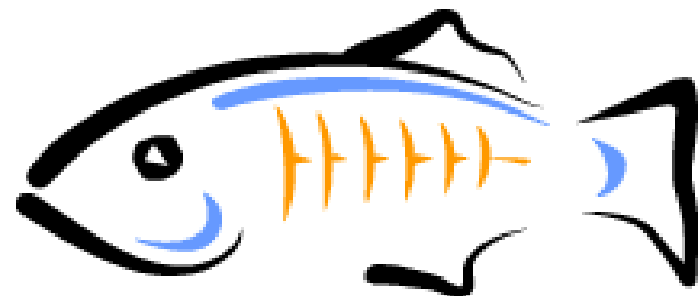
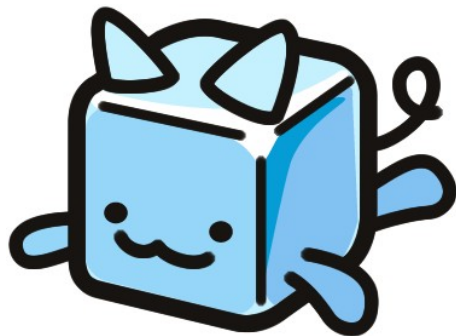
仕様の削減

古く使われなくなった API の排除

- > コンポーネントのオプション化
- > Java SE で検討されたプロセスと同様
- > 仕様削減候補：
 - > JAX-RPC(JAX-WS)
 - > EJB Entity Beans(Java Persistence API)
 - > JAXR
 - > JSR-88

進化するかんたん開発

- > すべての Web API でアノテーションを使用
- > web.xml の編集は不必要
- > 3rd ベンダーのライブラリを登録
- > 単純化されたパッケージ
- > RESTful Web サービス対応 (JAX-RS)
- > スクリプト言語のサポート



Java EE 6 に含まれる主な仕様

コンポーネント	JSR	仕様の承認状態
Java EE 6 Platform	JSR 316 http://jcp.org/en/jsr/detail?id=316	Public Draftの投票 02/23/2009
Servlet 3.0	JSR 315 http://jcp.org/en/jsr/detail?id=315	承認済み
EJB 3.1	JSR 318 http://jcp.org/en/jsr/detail?id=318	承認済み
Java Persistence 2.0	JSR 317 http://jcp.org/en/jsr/detail?id=317	承認済み
JSF 2.0	JSR 314 http://jcp.org/en/jsr/detail?id=314	承認済み
JAX-RS 1.0	JSR 311 http://jcp.org/en/jsr/detail?id=311	承認済み
Java EE Connector Architecture 1.6	JSR 322 http://jcp.org/en/jsr/detail?id=322	承認済み
Java Contexts and Dependency Injection (旧 Web Beans)	JSR 299 http://jcp.org/en/jsr/detail?id=299	承認済み
Concurrency Utilities for Java EE	JSR 236 http://jcp.org/en/jsr/detail?id=236	承認済み
Bean Validation	JSR 303 http://jcp.org/en/jsr/detail?id=303	承認済み

Web プロファイルに含まれる技術

オプション B を採用

コンポーネント

Servlet 3.0

JavaServer Pages (JSP) 2.2

Expression Language (EL) 2.2

Debugging Support for Other Languages (JSR-45) 1.0

Standard Tag Library for JavaServer Pages (JSTL) 1.2

JavaServer Faces (JSF) 2.0

Common Annotations for Java Platform (JSR-250) 1.1

Enterprise JavaBeans (EJB) 3.1 Lite

Java Transaction API (JTA) 1.1

Java Persistence API (JPA) 2.0

Servlet 3.0

- Servlet の新バージョン
- 非同期 (Async, Comet) のサポート
- web.xml のオプション化 (必須でない)
- セキュリティの拡張
 - > HttpServletRequest : login, logout
 - > HttpSession : logout
- アノテーション

```
import javax.servlet.http.annotation.*;  
@Servlet(urlMappings={"/foo"})  
public class SimpleSample {  
    ....  
}
```

JavaServer Faces 2.0

- > Ajax のサポート (Ajax Push, Comet は非対応)
- > ブックマーク可能な URL
- > エラーレポートの改善
- > 開発時間の軽減
- > カスタムコンポーネントの作成をかんたんに
- > 3rd ベンダーの JSF コンポーネントとの高い互換性を提供

EJB 3.1 の新機能

- > シングルトン Session Bean
 - > アプリケーションレベルで一度初期化
 - > @Startup アノテーションの追加
- > カレンダー表記のタイマー
 - > UNIX の cron に類似
 - > @Schedule アノテーションの追加
- > 自動生成タイマー
- > かんたんな非同期処理
 - > @Asynchronous アノテーションの追加

GlassFish v3 のご紹介

GlassFish v3

- Java EE 6 のリファレンス実装
 - > 軽量、高速、モジュール化
- Web 2.0 の実現に最適なコンテナ
 - > Java 言語、スクリプト言語 (JRuby 等) のサポート
 - > Java EE 6 のプロファイルに対応
- かんたんな操作
 - > 入手、インストール、管理、拡張、設定変更、開発
- Java EE も動作するアプリケーションサーバ
 - > Java EE 対応だけではない
 - > SOA の実現に最適



GlassFish v3 プレリユード

GlassFish v3 の一部機能を取前でリリース

- **かんたんインストール**
 - > GUI インストーラ付属
- **進化したアップデートセンター**
 - > 管理画面との統合
- **マイクロカーネルアーキテクチャ**
 - > OSGi 対応 / 高速起動 / リソース消費の軽減
- **Ruby on Rails のネイティブサポート**
 - > war へのアーカイブ必要無
- **無停止でアプリケーションの再配備**
 - > `asadmin redeploy --properties keepSessions=true`



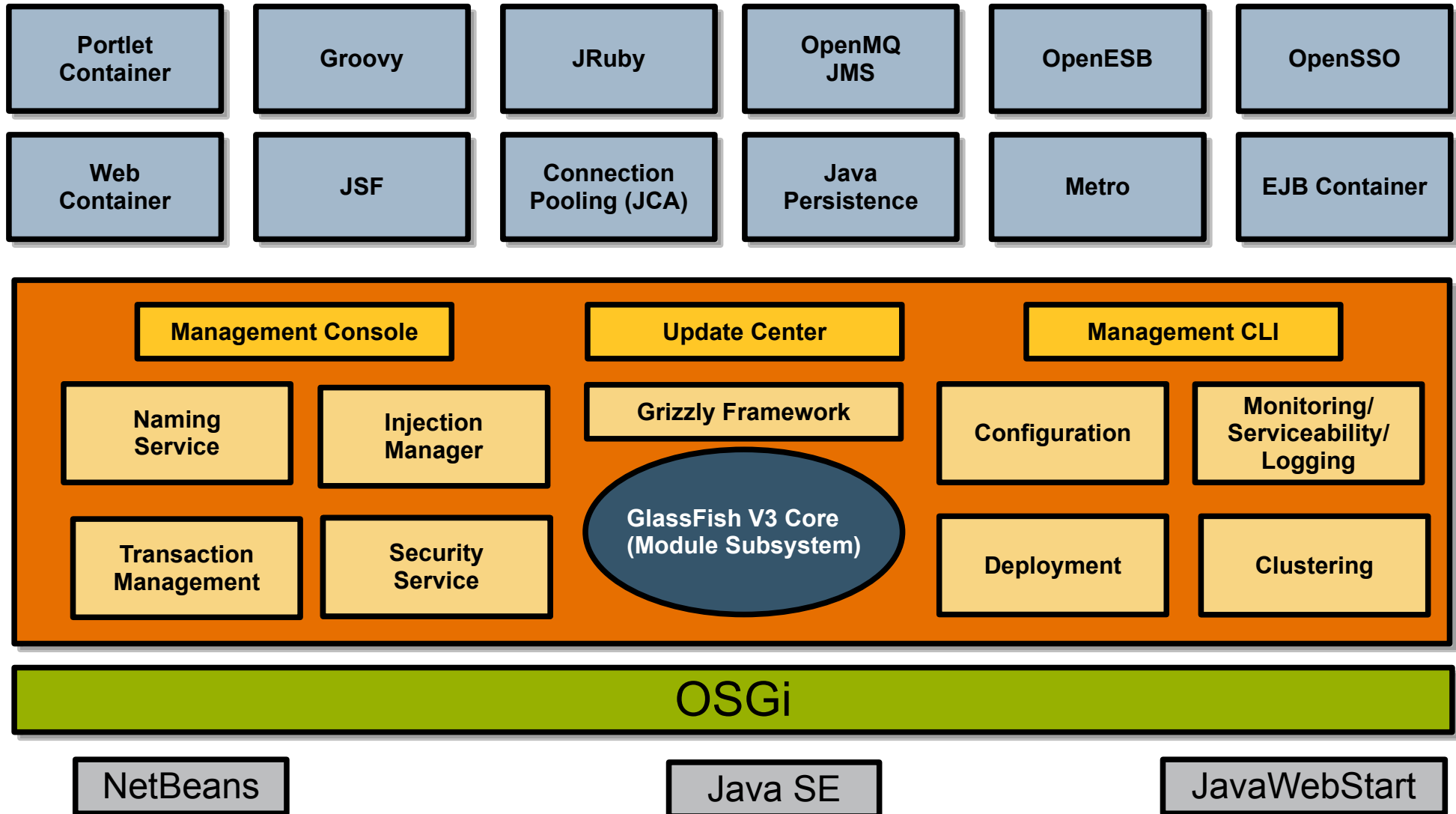
GlassFish v3 プレリュードリリース !!

GlassFish v3 の一部機能を取組みでリリース

- Sun の商用サポートを提供
- 埋め込み GlassFish API の提供
- サポートテクノロジー
 - > Java EE 5 ベース (一部 Java EE 6 を利用可能)
 - > Servlet 2.5
 - > Java Persistence API 1.0
 - > JAX-WS
 - > JavaServer Faces 1.2
 - > JavaServer Pages 2.1, JSTL 1.2
 - > Streaming API for XML (StAX)

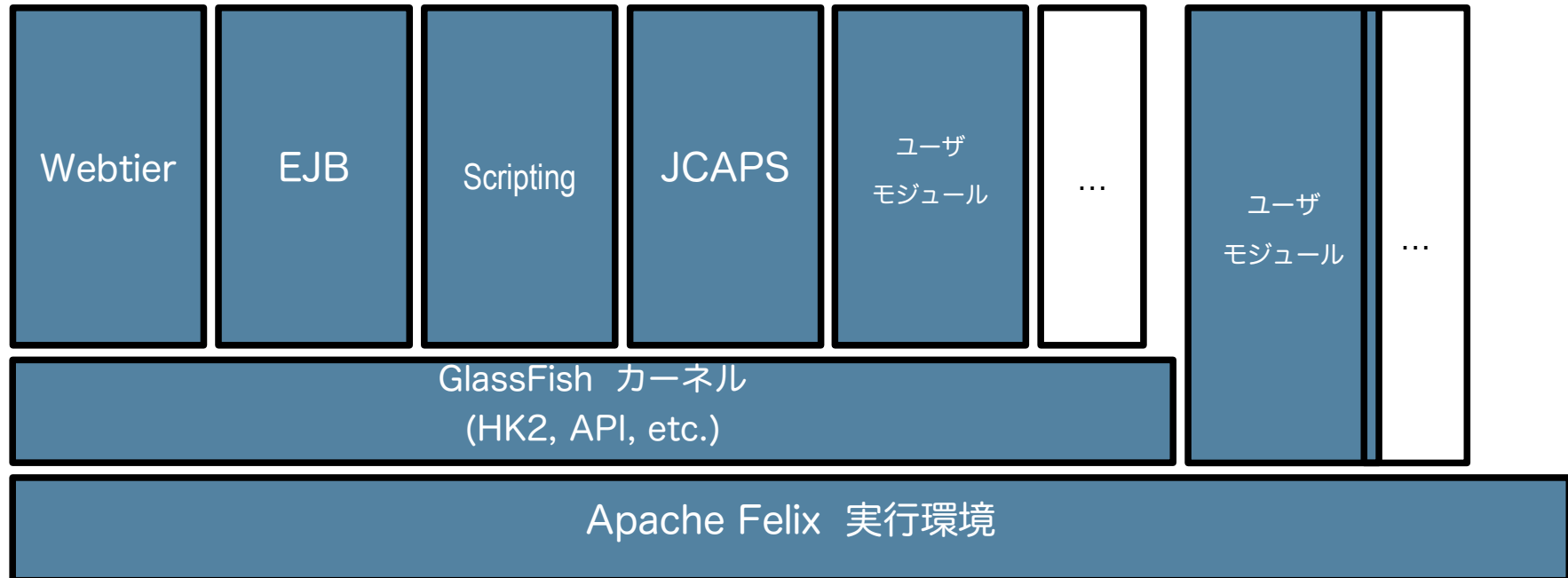


GlassFish v3 アーキテクチャ



GlassFish v3 実行環境

- GlassFish v3 は OSGi アプリケーション
 - > 独立したモジュール群から構成
 - > モジュールは OSGi コンテナで稼働 (Apache Felix)



起動時間の劇的な短縮

- 約 2 ～ 3 秒で起動
 - > 開発環境では起動、終了、 deploy 、 undeploy の速度が重要
- 使われてない機能を初期化しない
IIOP, JMS, web service, ...
- 開発環境・テスト環境で使いやすいものに

MacBook での実際の起動時間

Sun GlassFish Enterprise Server v3 Prelude startup time :

Felix(2751ms) startup services(1204ms) total(3955ms)

埋め込み GlassFish

- GlassFish をライブラリとして利用可能
 - > JVM の主人としてではなく
 - > GlassFish をインストールせずにモジュールを起動
- まだ携帯電話には載りません
- 内部的には異なる環境
 - > OSGi なし、モジュールなし、インストールディレクトリなし
 - > asadmin なし、JMX なし、自動 deploy なし
(勿論、必要があれば使えます)

例： `java -jar MyApplication.jar`

埋め込み GlassFish

- ひとつの jar にまとめられた GlassFish
- 管理ツールに代わるプログラム API を提供
 - > 起動 / 停止 / 設定 / 配備
- Maven プラグインも利用可能
 - > Unit テストも容易

```
GlassFish glassfish = new GlassFish();
glassfish.minimallyConfigure(8080);
GFApplication app = glassfish.deploy(new File("path/to/simple.war"));
...
app.undeploy();
glassfish.stop();
```

GlassFish v3 のまとめ

- モジュール化
 - マイクロモジュールカーネル、OSGi 対応
- 軽量、高速起動、スケーラブル
- 埋め込み可能 / 拡張可能
- JavaEE 6 対応 / 他フレームワークの利用
- スクリプト言語対応 (JRuby, Jython 等)
- 開発から本番環境まで
- ソースコードとドキュメントがオープン
 - 現在 Technology Preview 2 を利用可能



進化し続けるオープンソース アプリケーションサーバ：GlassFish

サン・マイクロシステムズ株式会社
ソフトウェア・ビジネス統括本部
寺田 佳央

<http://blogs.sun.com/yosshi>