



Sun Java Enterprise System 2005Q4 技術の概要

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-3583
2005 年 10 月

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

Sun, Sun Microsystems, docs.sun.com, AnswerBook, AnswerBook2 は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標、登録商標もしくは、サービスマークです。

サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK, OpenBoot, JLE は、サン・マイクロシステムズ株式会社の登録商標です。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは、OPEN LOOK のグラフィカル・ユーザインタフェースを実装するか、またはその他の方法で米国 Sun Microsystems 社との書面によるライセンス契約を遵守する、米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

U.S. Government Rights Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本製品が、外国為替および外国貿易管理法(外為法)に定められる戦略物資等(貨物または役務)に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

本製品に含まれる HG-MinchoL, HG-MinchoL-Sun, HG-PMinchoL-Sun, HG-GothicB, H G-GothicB-Sun, および HG-PGothicB-Sun は、株式会社リコーがリコービマジクス株式会社からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。HeiseiMin-W3H は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。©Copyright OMRON Co., Ltd. 1995-2000. All Rights Reserved. ©Copyright OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved.

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

「ATOK Server/ATOK12」に含まれる郵便番号辞書(7桁/5桁)は日本郵政公社が公開したデータを元に制作された物です(一部データの加工を行っています)。

「ATOK Server/ATOK12」に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド』に添付のものを使用しています。

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

原典: Sun Java Enterprise System 2005Q4 Technical Overview

Part No: 819-2330

Revision A



051218@13215



目次

はじめに	13
1 Java Enterprise System の概要	19
Java Enterprise System が必要な理由	19
Java Enterprise System コンポーネント	21
システムサービスコンポーネント	22
サービス品質コンポーネント	24
共有コンポーネント	26
Java Enterprise System での作業	27
Java Enterprise System ソリューションのライフサイクル	27
Java Enterprise System の導入シナリオ	30
この章の重要な用語	32
2 Java Enterprise System ソリューションアーキテクチャー	35
Java Enterprise System アーキテクチャーのフレームワーク	35
次元 1: インフラストラクチャーサービスの依存関係	37
次元 2: 論理層	42
次元 3: サービスの品質	45
アーキテクチャーの 3 つの次元の統合	49
Java Enterprise System ソリューションアーキテクチャーの例	50
エンタープライズ通信のシナリオ	50
シナリオ例の論理アーキテクチャー	51
シナリオ例の配備アーキテクチャー	52
この章の重要な用語	52

3	Java Enterprise System 統合機能	55
	Java Enterprise System 統合インストーラ	55
	既存のソフトウェアのチェック	56
	依存性の確認	56
	初期設定	57
	アンインストール	57
	統合されたアイデンティティサービスとセキュリティーサービス	57
	単一アイデンティティ	58
	認証とシングルサインオン	59
	承認	60
	この章の重要な用語	61
4	Java Enterprise System ソリューションのライフサイクルの作業	63
	ソリューションのライフサイクルの作業	63
	配備前	65
	配備	66
	配備設計	66
	配備実装	68
	配備後	71
	この章の重要な用語	71
5	参照リスト: Java Enterprise System コンポーネント	73
	システムサービスコンポーネントの説明	73
	Sun Java System Access Manager 7 2005Q4	74
	Sun Java System Application Server Enterprise Edition 8.1 2005Q4	75
	Sun Java System Calendar Server 6 2005Q4	75
	Sun Java System Directory Server 5 2005Q4	75
	Sun Java System Instant Messaging 7 2005Q4	76
	Sun Java System Message Queue 3.6 2005Q4	76
	Sun Java System Messaging Server 6 2005Q4	77
	Sun Java System Portal Server 6 2005Q4	77
	Sun Java System Service Registry 3 2005Q4	78
	Sun Java System Web Server 6.1 2005Q4	78
	サービス品質コンポーネントの説明	78
	可用性コンポーネントの説明	78
	アクセスコンポーネントの説明	79
	管理コンポーネントの説明	82

共有コンポーネント 83

索引 85

表目次

表 1-1	Java ES システムサービスコンポーネント	23
表 1-2	Java ES 可用性コンポーネント	25
表 1-3	Java ES アクセスコンポーネント	25
表 1-4	Java ES 管理コンポーネント	26
表 1-5	ライフサイクルの作業に関連する Java ES ユーザーカテゴリ	29
表 1-6	Java ES 導入シナリオに関する考慮事項	31
表 2-1	Java ES システムサービスコンポーネント間の関係	41
表 2-2	ソリューションアーキテクチャーに影響するサービス品質	46
表 2-3	サービス品質コンポーネントと影響を受けるシステム品質	47
表 2-4	ビジネス要件の要約: 通信のシナリオ	50

目次

図 1-1	分散型のエンタープライズアプリケーションに必要なサポート	20
図 1-2	Java ES コンポーネントのカテゴリ	22
図 1-3	ソリューションライフサイクルのフェーズとユーザーカテゴリ	28
図 2-1	ソリューションアーキテクチャーの次元	36
図 2-2	次元 1: インフラストラクチャーサービスレベル	38
図 2-3	Java ES システムサービスコンポーネント	40
図 2-4	次元 2: 分散型エンタープライズアプリケーションの論理層	42
図 2-5	Messaging Server: 層によるアーキテクチャーの例	45
図 2-6	Sun Cluster ノードによる可用性の設計	48
図 2-7	エンタープライズ通信のシナリオの論理アーキテクチャー	51
図 3-1	単一ユーザーエントリによる多数のサービスのサポート	58
図 3-2	認証順序	60
図 3-3	承認順序	61
図 4-1	ソリューションのライフサイクルの作業	64
図 4-2	配備シナリオの指定	65
図 4-3	配備シナリオに基づいた配備アーキテクチャーの作成	67

例目次

はじめに

『Java Enterprise System 技術の概要』では、Java Enterprise System の技術的および概念上の基礎を紹介します。また、Java Enterprise System のコンポーネント、アーキテクチャー、プロセス、および機能についても説明します。

ここでは、Java Enterprise System のマニュアルセットで使用される技術的な概念および用語を定義します。各章の「重要な用語」の項には重要な技術用語の説明があり、そこで Java Enterprise System の文脈での用語の使用方法を説明しています。

対象読者

『Java Enterprise System 技術の概要』は、Java Enterprise System に基づいてソフトウェアソリューションを設計、配備、または保守するユーザー向けに書かれています。したがって、ビジネスアナリスト、システム設計者、フィールドエンジニア、システム管理者などを含む幅広い読者が対象になります。

『Java Enterprise System 技術の概要』を読むには、次の各技術に関する一定の知識が必要になります。

- 一般的なネットワークの概念
- 認証および承認に関連するセキュリティの基礎
- Java 言語、Java 2 Standard Edition コンポーネント、Java 2 Enterprise Edition コンポーネント

Java Enterprise System のマニュアル セット

Java ES のマニュアルセットでは、配備計画およびシステムインストールについて説明します。システムマニュアルの URL は、<http://docs.sun.com/coll/1286.1> です。Java ES の概要を把握するには、次の表に記載された順番でマニュアルを参照してください。

表 P-1 Java Enterprise System のマニュアル

マニュアル名	内容
『Sun Java Enterprise System 2005Q4 リリースノート』	既知の問題など、Java ES に関する最新の情報が記載されています。これ以外に、コンポーネントごとにリリースノートがあります。
『Sun Java Enterprise System 2005Q4 ドキュメントロードマップ』	Java ES に関するすべてのマニュアルについて説明しています。システム全体に対するマニュアルと個々のコンポーネントに対するマニュアルの両方が含まれています。
『Sun Java Enterprise System 2005Q4 技術の概要』	Java ES の技術的および概念上の基礎を紹介しています。コンポーネント、アーキテクチャー、プロセス、および機能についても説明しています。
『Sun Java Enterprise System 2005Q4 配備計画ガイド』	Java ES に基づくエンタープライズ配備ソリューションの計画および設計について説明しています。配備計画および設計の基本的な概念と原則を示し、ソリューションのライフサイクルについて説明し、Java ES に基づくソリューションを計画する際に利用できる高度な例や方法を提供しています。
『Sun Java Enterprise System 2005Q4 インストール計画ガイド』	ハードウェア、オペレーティングシステム、ネットワークの各側面に関する Java ES 配備の実装仕様を作成する際に役立ちます。コンポーネントの依存関係など、インストールおよび設定計画時に解決すべき問題について説明しています。
『Sun Java Enterprise System 2005Q4 インストールガイド(UNIX 版)』	Java ES を Solaris オペレーティングシステムまたは Linux オペレーティングシステムへインストールする手順について解説しています。インストール後にコンポーネントを設定する方法やそれらが正しく機能しているか検証する方法についても説明しています。
『Sun Java Enterprise System 2005Q4 インストールリファレンス』	設定パラメータに関する追加情報、設定計画時に使用するべきワークシート、およびデフォルトのディレクトリやポート番号などのリファレンス情報を提供します。

表 P-1 Java Enterprise System のマニュアル (続き)

マニュアル名	内容
『Sun Java Enterprise System 2005Q1 配備実例集: 評価のシナリオ』	Java ES を 1 つのシステムへインストールし、一連のコアサービス、共有サービス、およびネットワークサービスを設定し、設定したサービスにアクセス可能なユーザーアカウントをセットアップする方法について説明しています。
『Sun Java Enterprise System 2005Q4 アップグレードガイド』	Solaris オペレーティングシステムまたは Linux オペレーティング環境での Java ES のアップグレード手順を示しています。
『Sun Java Enterprise System 用語集』	Java ES のマニュアル内で使用される用語を定義しています。

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用しません。

表 P-2 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	machine_name% su Password:
<i>aabbcc123</i>	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。

表 P-2 表記上の規則 (続き)

字体または記号	意味	例
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep `^#define \ XV_VERSION_STRING`

コード例は次のように表示されます。

- C シェル

```
machine_name% command y|n [filename]
```

- C シェルのスーパーユーザー

```
machine_name# command y|n [filename]
```

- Bourne シェルおよび Korn シェル

```
$ command y|n [filename]
```

- Bourne シェルおよび Korn シェルのスーパーユーザー

```
# command y|n [filename]
```

[] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

マニュアル、サポート、およびトレーニング

Sun のサービス	URL	内容
マニュアル	http://jp.sun.com/documentation/	PDF 文書および HTML 文書をダウンロードできます。
サポートおよび トレーニング	http://jp.sun.com/supporttraining/	技術サポート、パッチのダウンロード、および Sun のトレーニングコース情報を提供します。

第 1 章

Java Enterprise System の概要

Sun Java™ Enterprise System (Java ES) は、ネットワークを介して、またはインターネット環境で分散しているエンタープライズ版アプリケーションをサポートするために必要なサービスを提供する一連のソフトウェアコンポーネントです。このマニュアルでは、このようなアプリケーションを「分散型のエンタープライズアプリケーション」と呼びます。

Java Enterprise System は、Sun ソフトウェアのリリース、配信の方法論、およびビジネスと価格設定の戦略も実現しています。ただし、このマニュアルでは Java Enterprise System のソフトウェアコンポーネントとコンポーネントが提供するサービスに重点を置いています。

この章では、Java Enterprise System およびシステムの使用に関連する作業を紹介し、次のトピックが含まれます。

- 19 ページの「Java Enterprise System が必要な理由」
- 21 ページの「Java Enterprise System コンポーネント」
- 27 ページの「Java Enterprise System での作業」
- 32 ページの「この章の重要な用語」

Java Enterprise System が必要な理由

現在のビジネスでは、ネットワークまたはインターネット環境を介して分散する、高いレベルのパフォーマンス、可用性、セキュリティ、スケーラビリティ、および保守性を備えたソフトウェアソリューションが必要とされています。

Java Enterprise System では、このような分散型のエンタープライズアプリケーション、つまり、通常次のような特徴を持つアプリケーションをサポートするために必要なインフラストラクチャーサービスを提供します。

- 分散型: アプリケーションは、地理的に離れたサイトも含むネットワーク環境にわたって配備された対話型のソフトウェアコンポーネントで構成されています。環境内のさまざまなコンピュータ上で動作するこれらの分散型コンポーネントは、互い

に連携して動作することで、**エンドユーザー**およびその他のビジネスアプリケーションに対して特定のビジネス機能を提供します。

- **エンタープライズ版:** アプリケーションの適用範囲と規模が、本稼働環境またはインターネットサービスプロバイダのニーズを満たしています。このアプリケーションは通常、企業全体に及んでおり、多くの部門、オペレーション、およびプロセスを1つのソフトウェアシステムに統合します。アプリケーションは、パフォーマンス、可用性、セキュリティ、スケーラビリティ、および保守性に関する高度なサービス要件を満たす必要があります。

分散型のエンタープライズアプリケーションでは、分散型コンポーネントが互いに通信できるようにしたり、それらのコンポーネントの動作を調整したり、セキュリティ保護されたアクセスを実装したりする、基盤となる一連のインフラストラクチャーサービスが必要となります。さらに、これらのインフラストラクチャーサービスは、コンピュータやネットワークリストから成るハードウェア環境によってサポートされます。このハードウェア環境には、SPARC® と x86 (Intel および AMD) のハードウェアアーキテクチャーが含まれます。

次の図に、全体的な階層スキーマを示します。Java Enterprise System は基本的に、[図 1-1](#) に示す分散型インフラストラクチャーサービス層を提供します。ただし、Java Enterprise System サービスには、エンドユーザーがアクセス可能なサービスなど、アプリケーションレベルのサービスもいくつか含まれます。

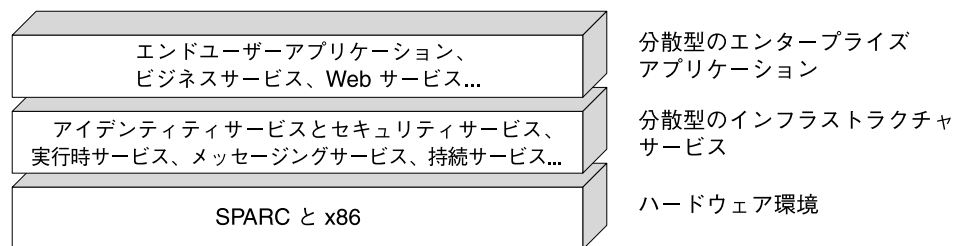


図 1-1 分散型のエンタープライズアプリケーションに必要なサポート

Java Enterprise System によって提供される主なサービスには、以下のものがあります。

- **ポータルサービス:** これらのサービスを使用すると、従業員、在宅勤務者、知識労働者、パートナ企業、仕入先、および顧客が、企業ネットワーク内外の任意の場所から企業のリソースにアクセスできます。これらのサービスは、ユーザーコミュニティに対する時間と場所を問わないアクセス、配信の個人用にカスタマイズされた統合、集約、セキュリティ、モバイルアクセス、および検索の機能を提供します。
- **通信サービスと共同作業サービス:** これらのサービスを使用すると、多様なユーザーコミュニティ間でセキュリティ保護された情報交換を行えます。具体的な機能には、ユーザーのビジネス環境で使用する、インスタントメッセージング、会議、カレンダーのスケジューリングなどのメッセージング機能によるリアルタイムの共同作業があります。

- ネットワークアイデンティティサービスとセキュリティサービス: グローバルベースのすべてのコミュニティー、アプリケーション、サービスの間で適切なアクセス制御が確実に適用されるようにすることで、企業の主要な情報資産のセキュリティと保護を向上させます。これらのサービスは、アイデンティティプロファイル、アクセス特権、およびアプリケーションとネットワークリソースの情報を格納、管理するリポジトリと連携して機能します。
- **Web** コンテナサービスとアプリケーションサービス: これらのサービスは、分散型コンポーネントが実行時に互いに通信することを可能にするほか、広範なサーバー、クライアント、およびデバイス用のアプリケーションの開発、配備、および管理をサポートします。これらのサービスは、J2EE™ (Java 2 Platform, Enterprise Edition) 技術に基づいています。

Java Enterprise System は、可用性、スケーラビリティ、保守性、およびその他のアプリケーションまたはシステムの品質を向上させるサービスも提供します。Java Enterprise System によって提供されるサービス品質の機能には、以下のものがあります。

- 可用性サービス: これらのサービスは、アプリケーションコンポーネントとそれらをサポートするインフラストラクチャーコンポーネントに対して、ほぼ連続的な可用性を提供します。
- アクセスサービス: これらのサービスは、Java Enterprise System サービスへのインターネットまたはブラウザベースのアクセスを提供します。
- 管理サービス: これらのサービスは、Java Enterprise System がサポートするアプリケーションのパフォーマンスを維持および調整する際に役立ちます。

ユーザーは、1 つまたは複数の Java Enterprise System サービスを配備できます。それらの各サービスには Java Enterprise System コンポーネントがいくつか含まれます。

Java Enterprise System コンポーネント

Java Enterprise System は、個別のソフトウェア製品やソフトウェアコンポーネントを単一のソフトウェアシステムとして統合化したものです。このシステムのコンポーネントに対するテストが実施されていて、各コンポーネント間の相互動作が保証されています。そうした統合は、次のようないくつかのシステムレベル機能によって実現されています。

- 共通に使用される一連の共有ライブラリですべてのコンポーネントの同期をとります。
- すべての Java ES コンポーネントは、単一のインストーラを使ってインストールされます。
- 統合された 1 つのユーザーアイデンティティおよびセキュリティ管理システムをすべての Java ES コンポーネントが共有できます。

これらの機能については、このマニュアルの後の章で説明します。ここでは、Java Enterprise System に統合されたさまざまなコンポーネントの説明に重点を置いています。これらのシステムコンポーネントは、次の図に示すように、3つの主要カテゴリに分類できます。

- システムサービスコンポーネント: 分散型のエンタープライズアプリケーションをサポートする主な Java Enterprise System インフラストラクチャーサービスを提供します。
- サービス品質コンポーネント: システムサービスコンポーネントおよび分散型アプリケーションコンポーネントの可用性、セキュリティ、スケーラビリティ、保守性、およびその他の要素を向上させます。
- 共有コンポーネント: 多くのシステムサービスコンポーネントおよびサービス品質コンポーネントを実行する環境を提供します。

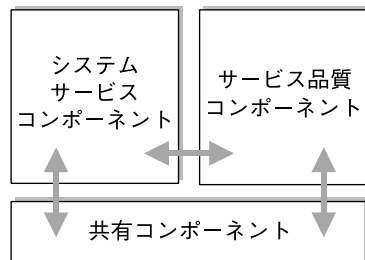


図 1-2 Java ES コンポーネントのカテゴリ

システムサービスコンポーネント

いくつかの Java ES コンポーネントが分散型ソフトウェアソリューションをサポートする主なサービスを提供します。これらのシステムサービスには、ポータルサービス、通信サービスと共同作業サービス、アイデンティティサービスとセキュリティサービス、Web コンテナサービス、および J2EE アプリケーションサービスが含まれます。

これらの分散型サービスを提供するシステムサービスコンポーネントとそれらが提供するサービスを簡単に説明したのが、次の表です。各システムサービスコンポーネントはマルチスレッド対応のサーバープロセスであり、多数のクライアントをサポートします。コンポーネントの詳細については、73 ページの「システムサービスコンポーネントの説明」を参照してください。

表 1-1 Java ES システムサービスコンポーネント

コンポーネント	提供されるシステムサービス
Sun Java System Access Manager	アクセス管理サービスおよびデジタルアイデンティティ管理サービスを提供します。アクセス管理サービスには、シングルサインオンを含む認証と、アプリケーションまたはサービス、あるいはその両方へのアクセスに対するルールに基づく承認が含まれます。管理サービスには、個々のユーザーアカウント、ロール、グループ、およびポリシーの集中管理が含まれます。
Sun Java System Application Server	セッション Beans、エンティティ Beans、メッセージ駆動型 Beans など、EJB (Enterprise JavaBeans™) コンポーネントの J2EE コンテナサービスを提供します。コンテナは、密接に結合された分散コンポーネント間の対話に必要なインフラストラクチャーサービスを提供し、Application Server を e-コマースアプリケーションおよび Web サービスを開発および実行するためのプラットフォームにします。Application Server は Web コンテナサービスも提供します。
Sun Java System Calendar Server	エンドユーザーおよびエンドユーザーグループにカレンダーサービスとスケジュールリングサービスを提供します。Calendar Server には、サーバーと対話するブラウザベースのクライアントが含まれています。
Sun Java System Directory Server	アイデンティティプロフィール (従業員、顧客、仕入先など)、ユーザーの信用情報 (公開鍵の証明書、パスワード、PIN 番号)、アクセス特権、アプリケーションリソース情報、ネットワークリソース情報などのイントラネット情報およびインターネット情報を格納および管理するための中央リポジトリを提供します。
Sun Java System Instant Messaging	インスタントメッセージング (チャット)、会議、アラート、ニュース、調査、ファイル転送など、エンドユーザー間における、セキュリティ保護されたリアルタイムの通信を提供します。このサービスには、ユーザーが現在オンラインであるかどうかと通知するプレゼンスマネージャー、およびブラウザからサーバーとの対話を行なうクライアントが含まれます。
Sun Java System Message Queue	緩やかに結合された分散型のコンポーネントやアプリケーション間における、信頼性の高い非同期のメッセージングを提供します。Message Queue は、JMS (Java Message Service) API 仕様を実装しているほか、さらにセキュリティ、スケーラビリティ、リモート管理などのエンタープライズ機能も備えています。
Sun Java System Messaging Server	電子メール、ファックス、ポケベル、音声、およびビデオをサポートする、セキュリティ保護された、信頼性の高い大容量の蓄積交換型メッセージングを提供します。Messaging Server は、複数のメッセージストアに同時にアクセスできるほか、一方的に送られてくる電子メールの受信拒否やウィルス攻撃の防止に役立つコンテンツフィルタリング機能を提供します。

表 1-1 Java ES システムサービスコンポーネント (続き)

コンポーネント	提供されるシステムサービス
Sun Java System Portal Server	ビジネスアプリケーションやビジネスサービスにアクセスするブラウザベースのクライアントに対し、コンテンツの集約や個人用のカスタマイズなど、主要なポータルサービスを提供します。Portal Server は、設定可能な検索エンジンも提供します。
Sun Java System Web Server	Java サブレットコンポーネントや JSP™ (JavaServer Pages™) コンポーネントなどの Java Web コンポーネントに対して、J2EE™ Web コンテナサービスを提供します。Web Server は、CGI スクリプトや Sun Java System Active Server Pages など、静的および動的な Web コンテンツを配信するためのその他の Web アプリケーション技術もサポートしています。
Sun Java System Service Registry	サービス指向アーキテクチャ (SOA) に対応した Web アプリケーションをサポートするレジストリおよびリポジトリを提供します。Service Registry は、Web サービスを登録および検出するための業界標準に加え、関連情報や事実、XML スキーマ、ビジネスプロセスルール、アクセス制御、バージョン管理といったアーティファクトなどを管理するための業界標準を実装しています。

サービス品質コンポーネント

Java Enterprise System には、表 1-1 に示したシステムサービスコンポーネントのほかに、システムサービスコンポーネントが提供するサービスの品質を高めるためのコンポーネントがいくつか含まれています。また、サービス品質コンポーネントを使えば、カスタム開発されたアプリケーションサービスも改善できます。サービス品質コンポーネントは次のカテゴリに分類されます。

- 可用性コンポーネント
- アクセスコンポーネント
- 管理コンポーネント

可用性コンポーネント

可用性コンポーネントは、システムサービスコンポーネントおよびカスタムアプリケーションサービスがほぼ連続的に稼動することを可能にします。Java Enterprise System に含まれる可用性コンポーネントおよびそれらのコンポーネントが提供するサービスを次の表に示します。コンポーネントの詳細については、78 ページの「可用性コンポーネントの説明」を参照してください。

表 1-2 Java ES 可用性コンポーネント

コンポーネント	提供される可用性サービス
Sun Cluster	Java Enterprise System の高可用性サービスとスケーラビリティサービス、Java Enterprise System インフラストラクチャーの最上部で実行されるアプリケーション、およびサービスとアプリケーションの両方が配備されるハードウェア環境を提供します。
High Availability Session Store	障害発生時でも、アプリケーションのデータ、特にセッション状態データを利用可能にするデータストアを提供します。

アクセスコンポーネント

アクセスコンポーネントは、システムサービスへのフロントエンドアクセスを可能にし、多くの場合、エンタープライズファイアウォールの外にあるインターネットからのセキュリティ保護されたアクセスを可能にします。そのようなアクセスを可能にすることに加えて、多くの場合、ルーティング機能も提供します。Java Enterprise System に含まれるアクセスコンポーネントおよびそれらのコンポーネントが提供するサービスを次の表に示します。コンポーネントの詳細については、79 ページの「[アクセスコンポーネントの説明](#)」を参照してください。

表 1-3 Java ES アクセスコンポーネント

コンポーネント	提供されるアクセスサービス
Sun Java System Directory Proxy Server	企業ファイアウォールの外からセキュリティサービスを Directory Server に対して提供します。Directory Proxy Server は、ディレクトリアクセス制御と複数の Directory Server インスタンスに対するルーティングを提供します。
Sun Java System Web Proxy Server	Web コンテンツをキャッシュ、フィルタリング、および配信する機能を、送信インターネット要求と受信インターネット要求の両方に対して提供します。
Sun Java System Communications Express	設定に応じて、Messaging Server、Calendar Server、および Directory Server への Web ベースのアクセスを提供します。
Sun Java System Portal Server, Secure Remote Access	企業ファイアウォールの外から、内部ポータルを含む Portal Server のコンテンツやサービスへの、セキュリティ保護されたインターネットアクセスを提供します。
Sun Java System Connector for Microsoft Outlook	Microsoft Outlook を使用するデスクトップクライアントに、Messaging Server と Calendar Server の両方に対するインタフェースを提供します。

管理コンポーネント

管理コンポーネントは、システムサービスのための、設定や監視などの管理機能を提供します。Java Enterprise System に含まれる管理コンポーネントおよびそれらのコンポーネントが提供するサービスを次の表に示します。コンポーネントの詳細については、[82 ページの「管理コンポーネントの説明」](#)を参照してください。

表 1-4 Java ES 管理コンポーネント

コンポーネント	提供される管理サービス
Sun Java System Administration Server (およびコンソール)	Directory Server および Messaging Server の設定および管理を可能にするグラフィカル管理ツールを提供します。
Sun Java System Directory Preparation Tool	Messaging Server および Calendar Server 用のユーザーのプロビジョニングに必要となるスキーマを使って Directory Server を設定するためのスクリプトを提供します。
Sun Java System Delegated Administrator	Messaging Server および Calendar Server が必要とするユーザー属性を Directory Server のユーザーエントリ内に格納するための、コマンド行ツールと GUI ツールの両方を提供します。

共有コンポーネント

Java Enterprise System には、多くのシステムサービスコンポーネントおよびサービス品質コンポーネントが依存する、ローカルにインストールされる共有ライブラリがいくつか含まれています。Java ES 共有コンポーネントは、同じホストコンピュータ上で稼働する Java ES コンポーネントにローカルサービスを提供します。

共有コンポーネントは、多くの場合、異なるオペレーティングシステム間の移植性を提供するために使用されます。Java Enterprise System 共有コンポーネントの例として、Java 2 Platform, Standard Edition (J2SE™ プラットフォーム)、Netscape Portable Runtime (NSPR)、Network Security Services (NSS)、Network Security Services for Java (JSS) などがあります。完全な一覧については、[83 ページの「共有コンポーネント」](#)を参照してください。

共有コンポーネントのインストールは、インストール対象のシステムサービスやサービス品質コンポーネントに応じて、Java ES インストーラによって自動的に行われます。

Java Enterprise System での作業

Java Enterprise System ソフトウェアに基づくビジネスソリューションの作成には、いくつかの標準的な作業が含まれます。それらの作業の範囲や難易度は、Java Enterprise System 導入時の出発点や、作成および配備しようとしているソリューションの性質によって異なります。

ここでは、Java Enterprise System での作業における 2 つの側面、つまり、Java Enterprise System ソリューションのライフサイクルと通常使用される各種導入シナリオについて説明します。

Java Enterprise System ソリューションのライフサイクル

Java ES ソフトウェアに基づくビジネスソリューションの作成に伴う作業は、[図 1-3](#) に示すように、いくつかのフェーズに分けられます。この図は、各種作業を通常実行する Java Enterprise System ユーザーのカテゴリも示しています。

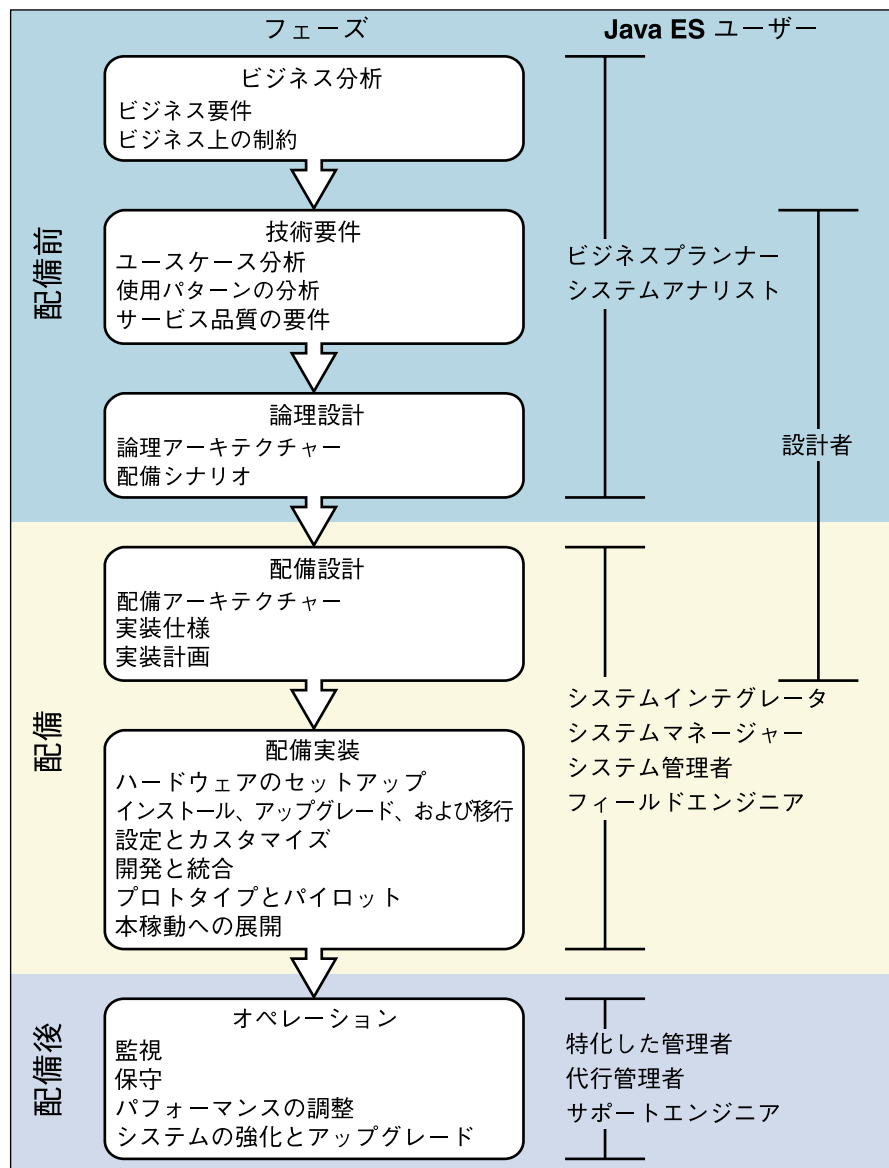


図 1-3 ソリューションライフサイクルのフェーズとユーザーカテゴリ

図 1-3 に示すライフサイクルのフェーズは、次の一般的なグループに分類できます。

- 配備前: このフェーズでは、ビジネスのニーズを配備シナリオ、つまり論理アーキテクチャーと一連のサービス品質の要件に変換します。配備シナリオは、配備アーキテクチャーを設計するために使用する仕様としての役割を果たします。

- **配備:** このフェーズでは、配備シナリオに基づいて配備アーキテクチャーを作成します。このアーキテクチャーをプロジェクト承認および予算設定の基礎として使用できます。この配備アーキテクチャーは、ソフトウェアソリューションを本稼働環境に配備 (構築、テスト、および展開) するために必要な詳細情報が含まれる実装仕様の基礎にもなります。
- **配備後:** オペレーションフェーズでは、配備されたソリューションを本稼働条件下で実行し、パフォーマンスを監視および最適化します。また、配備されたソリューションを必要に応じてアップグレードし、新しい機能を組み入れます。

図 1-3 に示すソリューションのライフサイクルおよび各フェーズの作業については、第 4 章で詳しく説明します。

図 1-3 は、ライフサイクルの各フェーズに示された作業を通常実行する Java ES ユーザーを示しています。Java ES で作業を実行する場合、実行するジョブは図 1-3 に示す 1 つ以上のユーザーカテゴリに該当します。各カテゴリのユーザーのスキルとバックグラウンドは、次の表のとおりです。

表 1-5 ライフサイクルの作業に関連する Java ES ユーザーカテゴリ

ユーザー	スキルとバックグラウンド	フェーズ
ビジネスプランナ システムアナリスト	専門的ではなくても、一般的な技術知識を持っている。 企業の戦略上の方向性を理解している。 ビジネスのプロセス、目的、要件を理解している。	ビジネス分析 技術要件 論理設計
設計者	専門的である。 配備アーキテクチャーの幅広い知識を持つ。 最新の技術に精通している。 ビジネスの要件および制約を理解している。	論理設計 配備設計
システムインテグレータ フィールドエンジニア システム管理者 システムマネージャー	専門的である。 IT 環境に精通している。 分散型のソフトウェアソリューションを実装した経験がある。 ネットワークのアーキテクチャー、プロトコル、デバイス、セキュリティの知識を持つ。 スクリプト言語およびプログラミング言語の知識を持つ。	配備設計 配備実装

表 1-5 ライフサイクルの作業に関連する Java ES ユーザーカテゴリ (続き)

ユーザー	スキルとバックグラウンド	フェーズ
特化したシステム管理者 代行管理者 サポートエンジニア	<p>特化した技術知識または製品知識を持つ。</p> <p>ハードウェア、プラットフォーム、ディレクトリ、およびデータベースに精通している。</p> <p>ソフトウェアの監視、トラブルシューティング、およびアップグレードに熟練している。</p> <p>オペレーティングシステムプラットフォームのシステムの管理の知識を持つ。</p>	オペレーション

Java Enterprise System の導入シナリオ

Java ES の導入につながるビジネスのニーズは多様です。ただし、ほとんどすべての Java ES 配備の高い目標は、次のいずれかの導入シナリオに該当します。

- **新しいシステム:** 既存のソフトウェアシステムなしで始め、新しいビジネスソリューションをサポートするために Java Enterprise System ソフトウェアを配備します。
- **強化:** 既存の情報技術 (IT) インフラストラクチャーで始め、そのシステムの一部、大部分、または全部を Java ES ソフトウェアで置き換えます。通常は、システムまたはサブシステムが複雑すぎるか、制約が多すぎるか、維持するのに高価であるという理由でシステムやサブシステムを置き換えます。たとえば、セキュリティの向上、高い可用性、スケーラビリティの向上、柔軟性の向上、複雑さの解消、追加機能 (シングルサインオンなど)、または IT リソースの有効利用が必要であるなどです。つまり、既存のシステムで得られるよりも優れた投資効果を望んでいるということです。
- **拡張:** 既存の IT インフラストラクチャーで始め、現在システムに含まれない Java Enterprise System ソフトウェアを配備します。通常、新しいビジネスのニーズに対応するために、この方法でソフトウェアシステムを拡張します。Java ES ポータルによる既存のサービスの個人用にカスタマイズされた集約や既存のサービスに対する Java 認証および承認などの新しい機能が必要になる場合があります。
- **アップグレード:** Java Enterprise System の以前のバージョンまたは Java Enterprise System よりも前の Sun 製品から構成される IT インフラストラクチャーで始め、Java Enterprise System コンポーネントの最新のバージョンにアップグレードします。

各導入シナリオには、それぞれ考慮しなければならない点と克服しなければならない点があります。ユーザーの状況に当てはまる導入シナリオがどれであるかにかかわらず、図 1-3 に示すソリューションのライフサイクルプロセスが適用されます。ただし、導入シナリオによって、ライフサイクルの各フェーズで対処する必要のある問題や投資する必要のあるリソースが異なります。

一般に、導入シナリオには、次に示す考慮事項が程度の差はありますが適用されます。

- **移行:** 既存のインフラストラクチャーを新しいソフトウェアで強化またはアップグレードするには、多くの場合、既存のシステムから新しいシステムにデータを移行する必要があります。データは、設定情報、ユーザー情報、アプリケーションの情報などになります。新しいプログラミングインタフェースのために、ビジネスロジックやプレゼンテーションロジックを移行する必要がある場合もあります。
- **統合:** 新しいソフトウェアを既存のシステムに追加したり、既存のソフトウェアサブシステムを置き換えたりするには、多くの場合、新しいソフトウェアコンポーネントを残りのサブシステムに統合する必要があります。統合には、新しいインタフェースレイヤーの開発、J2EE コネクタまたはリソースアダプタの使用、既存のコンポーネントの再設定、データ変換スキーマの実装などが含まれます。
- **トレーニング:** インフラストラクチャーの変更はほとんどの場合、IT 手順やスキルセットの変更を意味します。IT 部門は、Java Enterprise System テクノロジをサポートするための、新しいスキルの習得、または古いスキルの移行のために十分な時間を確保できる必要があります。
- **ハードウェア:** 既存のシステムまたはサブシステムを置き換えたり強化したりする場合、ビジネス上の制約により、既存のハードウェアを再利用しなければならない可能性があります。導入シナリオによっては、ハードウェアリソースが重要な要素になります。

次の表は、Java ES の各導入シナリオに該当する考慮事項の性質を要約しています。

表 1-6 Java ES 導入シナリオに関する考慮事項

導入シナリオ	移行	統合	トレーニング	ハードウェア
新しいシステム	問題なし	新しいコンポーネントの統合は比較的簡単	通常かなりの考慮が必要	機器のコストと労力のコストのバランス ¹
強化	重要な考慮事項になる場合がある	新しいコンポーネントを既存のシステムと統合する必要がある	かなりの考慮が必要な場合がある	既存の機器によるかなりの制約がある場合がある
拡張	通常は問題なし	新しいコンポーネントを既存のシステムと統合する必要がある場合がある	かなりの考慮が必要な場合がある	一般に、新しいシステムと同じトレードオフの新しいハードウェアが必要になる
アップグレード	かなりの考慮が必要な場合がある	アップグレードされたコンポーネントの統合は比較的簡単	比較的考慮事項が少ない	比較的考慮事項が少ない

¹ 少数の強力なコンピュータを使用すると、通常、機器のコストは増えますが、必要な IT リソースは少なくなります。多数の小型のコンピュータを使用すると、通常、機器のコストは少なくなります。より多くの IT リソースが必要になります。

この章の重要な用語

この節では、この章で使用されている重要な技術用語について説明します。ここでは、用語間の関係や Java Enterprise System の文脈でどのように使用されているかの説明に重点を置いています。

導入シナリオ	Java Enterprise System ソフトウェアを配備する総合的な理由。着手するソフトウェアシステムや達成すべき目標を明らかにします。Java Enterprise System には基本的な導入シナリオが 4 つあります。新しいシステム、置換、拡張、およびアップグレードです。
コンポーネント	分散型アプリケーションを構築するときの基本となるソフトウェアロジックの単位。コンポーネントは、Java Enterprise System に含まれるシステムコンポーネント、カスタム開発されたアプリケーションコンポーネントのいずれかになります。通常、アプリケーションコンポーネントは、CORBA や J2EE™ などの分散型コンポーネントモデルに準拠していて、いくつかの特定のコンピューティング機能を実行します。これらのコンポーネントを単独でまたは複数を組み合わせて、ビジネスサービスを提供します。また、これらを Web サービスとしてカプセル化することもできます。
分散型のエンタープライズアプリケーション	そのロジックがネットワーク環境またはインターネット環境に及んでおり (分散の側面)、かつその適用範囲と規模が本稼働環境またはサービスプロバイダのニーズを満たしている (エンタープライズの側面) アプリケーション。
エンドユーザー	分散型アプリケーションを利用するユーザー。エンドユーザーは通常、インターネットブラウザやモバイルデバイス GUI などのグラフィカルユーザーインターフェースを使用します。アプリケーションがサポートする同時接続エンドユーザーの数は、そのアプリケーションの配備アーキテクチャーを決定する重要な要因の 1 つになります。
サービス	1 つ以上のクライアントに対して実行されるソフトウェア機能。この機能は、メモリー管理など下位レベルのものであることも、信用調査ビジネスサービスなどの上位レベルのものであることも考えられます。上位レベルのサービスは、個々のサービスのファミリによって構成されることがあります。サービスには、ローカルクライアントで利用可能なローカルサービスと、リモートクライアントで利用可能な分散型サービスがあります。
サービス品質コンポーネント	Java Enterprise System に含まれる何種類かのシステムコンポーネントのうちの 1 つ。これらのコンポーネントは、システムサービスコンポーネントおよび分散型アプリケーションコンポーネントの可用性、セキュリティ、スケーラビリティ、保守性、およびその他の品質を向上させます。

共有コンポーネント	Java Enterprise System に含まれる何種類かのシステムコンポーネントのうちの一つ。共有コンポーネントは、通常、他のシステムコンポーネントにローカルサービスを提供するライブラリです。これに対し、システムサービスコンポーネントは、ほかのシステムコンポーネント(またはアプリケーションコンポーネント)に分散型インフラストラクチャーサービスを提供します。
システムコンポーネント	Java Enterprise System に含まれていて、Java Enterprise System インストーラによってインストールされるソフトウェアパッケージまたは一連のパッケージ。システムコンポーネントには、分散型インフラストラクチャーサービスを提供するシステムサービスコンポーネント、アクセスサービスや管理サービスを提供することでシステムサービスコンポーネントをサポートするサービス品質コンポーネント、およびほかのシステムコンポーネントにローカルサービスを提供する共有コンポーネントなど、いくつかの種類が存在します。
システムサービス	Java Enterprise System によって提供される固有の機能を定義する、1つ以上の分散型サービス。システムサービスは通常、いくつかのサービス品質コンポーネント、またはいくつかの共有コンポーネント、あるいはその両方のサポートを必要とします。
システムサービスコンポーネント	Java Enterprise System に含まれる何種類かのシステムコンポーネントのうちの一つ。システムサービスコンポーネントは、Java Enterprise System の主要インフラストラクチャーサービスを提供します。これには、ポータルサービス、通信サービスと共同作業サービス、アイデンティティサービスとセキュリティサービス、Web サービスとアプリケーションサービス、および可用性サービスが含まれます。

第 2 章

Java Enterprise System ソリューションアーキテクチャー

この章では、Java Enterprise System (Java ES) ソリューションの基礎となるアーキテクチャー概念の概要について説明します。この章では、Java ES コンポーネントである、システムサービスコンポーネントとサービス品質コンポーネントの両方を使用してどのように分散型エンタープライズソリューションをサポートするかを示します。

Java ES ソリューションアーキテクチャーには 2 つの側面があります。論理アーキテクチャーと配備アーキテクチャーです。論理アーキテクチャーは、ソリューションの論理的な構築ブロック (ソフトウェアコンポーネント) 間の対話を示します。配備アーキテクチャーは、論理アーキテクチャーから物理的なコンピューティング環境へのマッピングを示します。Java ES コンポーネントは、論理アーキテクチャーと配備アーキテクチャーの両方で重要な役割を果たします。

この章では、Java ES ソリューションのアーキテクチャーの設計のためのアーキテクチャーのフレームワークについて説明し、その後そのアーキテクチャーのフレームワークに基づいたソリューションアーキテクチャーの例を示します。

この章で説明する内容は、次のとおりです。

- 35 ページの「Java Enterprise System アーキテクチャーのフレームワーク」
- 50 ページの「Java Enterprise System ソリューションアーキテクチャーの例」
- 52 ページの「この章の重要な用語」

Java Enterprise System アーキテクチャーのフレームワーク

Java ES コンポーネントは、分散型エンタープライズ版ソフトウェアソリューションの配備をサポートします。

ビジネス要件によって課されるパフォーマンス、可用性、セキュリティー、スケーラビリティ、および保守性のレベルで必要な機能を実現するには、これらのソフトウェアソリューションを適切に設計する必要があります。

分散型エンタープライズ版ソフトウェアソリューションの設計には、アーキテクチャーの次元の多くが関係します。それらの次元は、そのようなシステムの構築に使用される多数のソフトウェアコンポーネント間の対話をさまざまな観点から見ることができる観点を表します。特に、分散型システムの設計にはアーキテクチャーの次の3つの次元が関係します。

- **インフラストラクチャーサービスの依存関係:** この次元では、分散型ソリューションのサポートにおけるシステムサービスコンポーネント (22 ページの「システムサービスコンポーネント」を参照) の役割に重点を置いています。
- **論理層:** この次元では、ソリューションコンポーネントをネットワークまたはインターネット環境に配備するためのソリューションコンポーネントの論理的および物理的な独立性に重点を置いています。
- **サービスの品質:** この次元では、サービス品質コンポーネント (24 ページの「サービス品質コンポーネント」を参照) の役割も含め、可用性、セキュリティー、スケーラビリティ、保守性といったサービス品質の要件をどのようにして実現するかに重点を置いています。

ソリューションアーキテクチャーのこれらの3つの次元を次の図に示します。

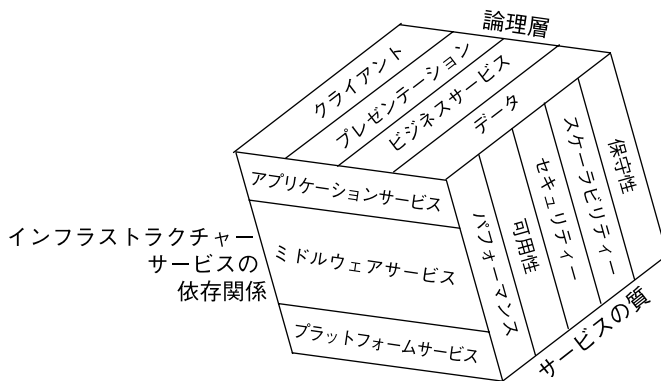


図 2-1 ソリューションアーキテクチャーの次元

これらの3つの次元によって、ソフトウェアソリューションに必要なサービス機能とサービス品質の実現に必要な、アプリケーションコンポーネントとインフラストラクチャーコンポーネントの両方の、ソフトウェアコンポーネント間の関係を統合する単一のフレームワークを表します。

次の各項では、3つの各次元を個別に説明し、その後、3つの次元を1つのフレームワークに統合して説明します。

次元 1: インフラストラクチャーサービスの依存関係

分散型のエンタープライズアプリケーションの対話型ソフトウェアコンポーネントには、基本となるインフラストラクチャーサービスのセットが必要です。これに基づいて、分散しているコンポーネント間で相互に通信したり、それぞれの動作を調整したり、セキュリティ保護されたアクセスを実装することなどが可能になります。ここでは、これらのインフラストラクチャーサービスを提供するためにいくつかの Java ES コンポーネントが果たす主な役割について説明します。

インフラストラクチャーサービスレベル

分散型のソフトウェアシステムを設計する場合、そのほとんどがカスタム開発コンポーネントで構成されるか、または追加設定の必要ない Java ES コンポーネントで構成されるかにかかわらず、多数のインフラストラクチャーサービスを組み込む必要があります。これらのサービスは、多数のレベルで機能します。

ソリューションアーキテクチャーのインフラストラクチャーサービスの依存関係の次元を、[図 2-2](#) に示します。この図に示されているレベルは、[図 1-1](#) のインフラストラクチャーサービス層を詳細化したものです。

[図 2-2](#) のサービス階層とサービス間の依存関係が、ソリューションの論理アーキテクチャーの重要な次元を構成します。これらのインフラストラクチャーサービスは、Java ES システムサービスコンポーネント ([22 ページの「システムサービスコンポーネント」](#)を参照) の役割を理解するための概念上の基礎になります。

一般的に、[図 2-2](#) に示したサービスは、大きく 3 つのグループに分けられます。下位レベルのプラットフォームサービス、上位レベルのアプリケーションサービス、およびミドルウェアサービスのグループです。なお、ミドルウェアサービスという名前は、ほかの 2 つのグループの間にあることから付けられたものです。

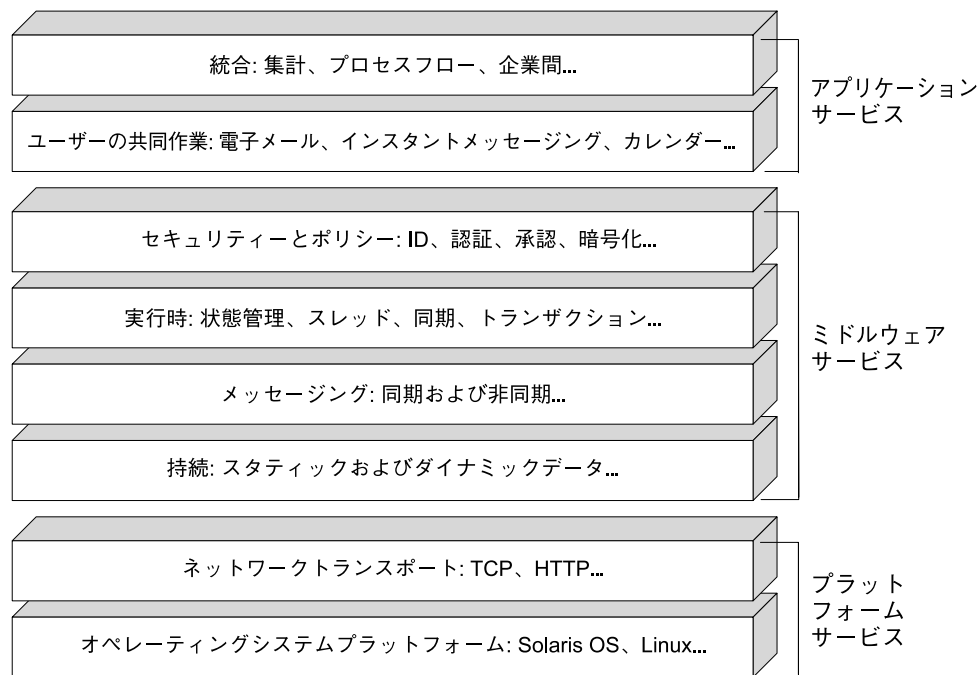


図 2-2 次元 1: インフラストラクチャーサービスレベル

次の各段落では、さまざまなインフラストラクチャーサービスレベルについて説明し、関連する場合には Java プログラミング言語のアーチファクトの参考情報も示します。図 2-2 に示された各サービスレベルについて、最下位レベルから順に説明します。

- **オペレーティングシステムプラットフォーム:** コンピュータ上で実行されるすべてのプロセスに対し、基本的なサポートを提供します。オペレーティングシステム (Solaris™ オペレーティングシステム、Linux、Microsoft Windows など) は、物理デバイスのほかに、メモリー、スレッド、および JVM™ (Java Virtual Machine) マシンのサポートに必要なその他のリソースを管理します。
- **ネットワークトランスポート:** 異なるコンピュータ上で実行されている分散型アプリケーションコンポーネント間での通信に必要な、基本的なネットワークサポートを提供します。これらのサービスには、TCP や HTTP などのプロトコルに対するサポートも含まれます。上位レベルのその他の通信プロトコル(「メッセージングレベル」を参照)は、これらの基本的なトランスポートサービスに依存しています。
- **持続:** 静的データ (ユーザー、ディレクトリ、設定情報など) と動的アプリケーションデータ (頻繁に更新される情報) の両方に対するアクセスや格納に必要なサポートを提供します。
- **メッセージング:** アプリケーションコンポーネント間の同期および非同期の通信に対するサポートを提供します。同期メッセージングでは、メッセージをリアルタイムで送受信します。これには、J2EE コンポーネント間のリモートメソッドの呼び

出し (RMI) や Web サービスとの SOAP 対話も含まれます。非同期メッセージングの通信では、直後に受信するコンシューマの準備状況に関係なく、メッセージを送信します。JMS (Java Message Service) や ebXML などの非同期メッセージングの仕様では、信頼性の保証およびその他のメッセージング方式をサポートします。

- **実行時: J2EE モデルや CORBA モデルなどの分散型のコンポーネントモデルで必要となるサポートを提供します。** 実行時サービスには、密接に結合された分散型コンポーネントに必要なリモートメソッドの呼び出しの他に、コンポーネントの状態 (ライフサイクル) の管理、スレッドプールの管理、同期 (相互排他ロック)、持続サービス、分散するトランザクションの監視、分散する例外の処理などが含まれます。J2EE 環境では、これらの実行時サービスはアプリケーションサーバーまたは Web サーバーの EJB™、Web、およびメッセージ駆動型 Bean コンテナから提供されます。
- **セキュリティーとポリシー: アプリケーションリソースへのセキュリティー保護されたアクセスに必要なサポートを提供します。** これらのサービスには、グループまたはロールに基づく分散型リソースへのアクセスを制御するポリシーのサポートや、**シングルサインオン** 機能が含まれます。シングルサインオンを使用すると、ある分散型システム内の 1 つのサービスに対するユーザー認証を、同じシステム内の他のサービス (J2EE コンポーネント、ビジネスサービス、Web サービスなど) に自動的に適用できます。
- **ユーザーの共同作業: ユーザー間の直接通信およびエンタープライズ内とインターネット環境内でのユーザー間の共同作業に対するサポートで重要な役割を果たすサービスを提供します。** これらのサービスは、一般的にスタンドアロンサーバー (電子メールサーバーやカレンダーサーバーなど) から提供されるアプリケーションレベルのビジネスサービスです。
- **統合: 既存のビジネスサービスを集約するサービスを提供します。** ポータルと同様にサービスにアクセスするための共通インタフェースを提供するか、これらのサービスを本稼働ワークフロー内で調整するプロセスエンジンを使用し、統合することによって行います。統合は、異なる企業間における企業間 (B2B) 統合として行なわれることもあります。

図 2-2 に示したサービスレベルは、最下位レベルのオペレーティングシステムサービスから最上位レベルのアプリケーションサービスや統合サービスまで、さまざまなインフラストラクチャーサービス間の一般的な依存関係を反映しています。通常、各サービスはその下にあるサービスに依存し、その上にあるサービスをサポートします。

ただし、図 2-2 は、インフラストラクチャーサービスの厳密な階層を表しているわけではありません。上位レベルのサービスは、中間のレベルに依存せずに、下位レベルのサービスと直接対話することができます。たとえば、一部の実行時サービスは、中間にあるサービスレベルを介さずに、プラットフォームサービスに直接依存する場合があります。さらに、監視サービスや管理サービスなどのその他のサービスレベルも、ここでの概念的な説明に含まれることがあります。

Java Enterprise System インフラストラクチャーサービスコンポーネント

Java ES コンポーネントは、図 2-2 に示した分散型インフラストラクチャーサービスレベルを実装したものです。Java ES システムサービスコンポーネントのさまざまなレベル内における位置関係を、図 2-3 に示します。

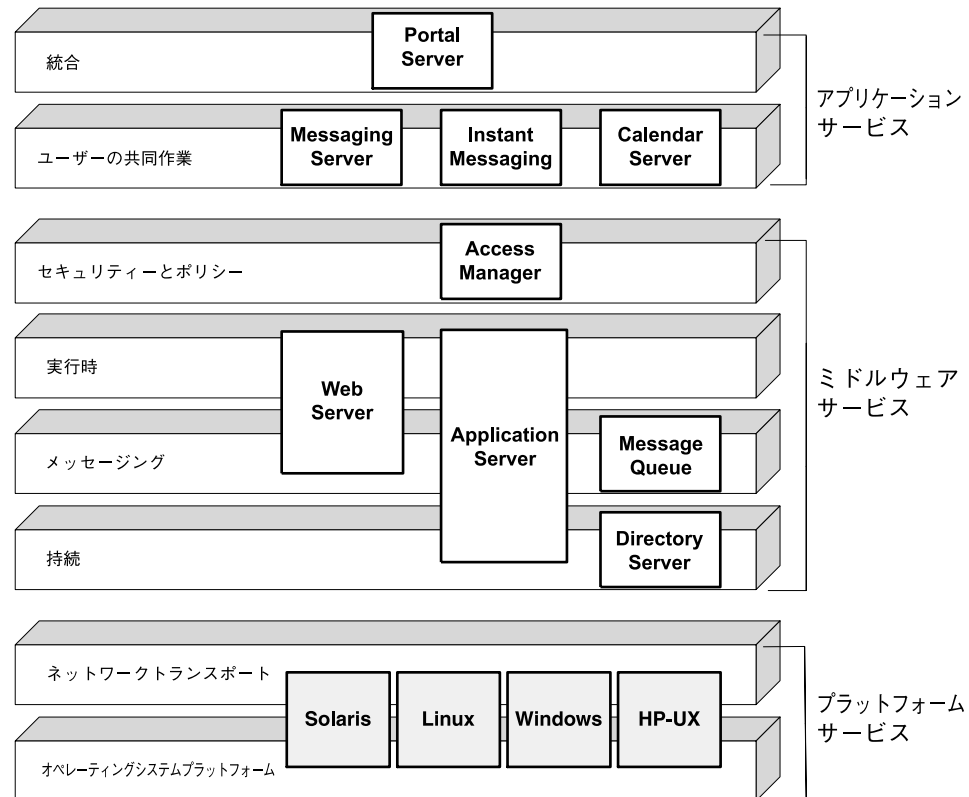


図 2-3 Java ES システムサービスコンポーネント

注 - 図 2-3 に示したオペレーティングシステムプラットフォームは正式には Java Enterprise System の一部ではありませんが、Java ES コンポーネントをサポートするオペレーティングシステムプラットフォームを示すために、この図に含めてあります。

Java Enterprise System インフラストラクチャーサービスの依存関係

一般に、図 2-3 に示した各 Java ES システムサービスコンポーネントは、インフラストラクチャー内でその下にあるコンポーネントに依存し、その上にあるコンポーネントをサポートします。それらの依存関係とサポートの関係は、論理アーキテクチャーを設計する上で重要な要素です。

表 2-1 に、Java ES システムサービスコンポーネント間の具体的な関係を示します。この表では図 2-3 と同様に、最上位から順に記載しています。

表 2-1 Java ES システムサービスコンポーネント間の関係

コンポーネント	依存するサーバー	サポートするサーバー
Portal Server	Application Server または Web Server Access Manager Directory Server 対応するチャネルを使用するように設定されている場合 Calendar Server Messaging Server Instant Messaging	
Messaging Server	Directory Server Access Manager (シングルサインオン用)	Calendar Server (電子メール通知用) Portal Server (メッセージングチャネル用)
Instant Messaging	Directory Server Access Manager (シングルサインオン用)	Portal Server (インスタントメッセージングチャネル用)
Calendar Server	Directory Server Messaging Server (電子メール通知サービス用) Access Manager (シングルサインオン用)	Portal Server (カレンダーチャネル用)
Access Manager	Application Server または Web Server Directory Server	Portal Server シングルサインオン用に設定されている場合: Calendar Server Messaging Server Instant Messaging
Application Server	Message Queue Directory Server (管理オブジェクト用)	Portal Server Access Manager
Message Queue	Directory Server (管理オブジェクト用)	Application Server

表 2-1 Java ES システムサービスコンポーネント間の関係 (続き)

コンポーネント	依存するサーバー	サポートするサーバー
Web Server	Access Manager (アクセス制御用)	Portal Server Access Manager
Directory Server	なし	Portal Server Calendar Server Messaging Server Instant Messaging Access Manager
Service Registry	なし	Application Server に基づくコンポーネント

次元 2: 論理層

分散型エンタープライズアプリケーションの相互に作用するソフトウェアコンポーネントは、いくつかの論理層に存在するとみなすことができます。これらの層は、提供するサービスの性質に基づいて、ソフトウェアコンポーネントの論理的および物理的な独立性を表しています。

ソリューションアーキテクチャーの論理層の次元を、次の図に示します。

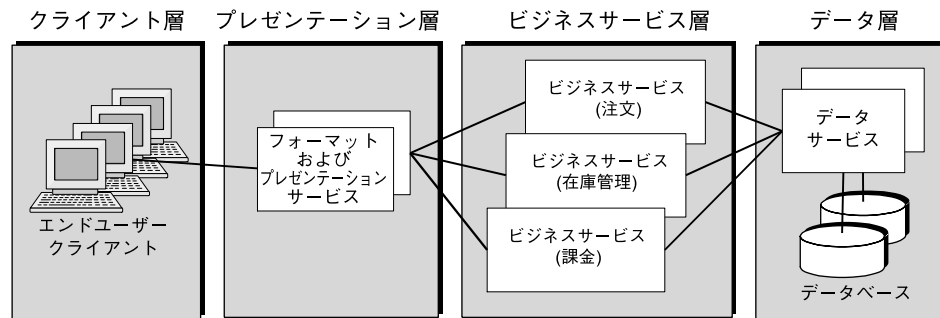


図 2-4 次元 2: 分散型エンタープライズアプリケーションの論理層

論理層アーキテクチャーは基本的に、図 1-1 の分散型エンタープライズアプリケーション層を表します。37 ページの「インフラストラクチャーサービスレベル」で説明した Java ES システムサービスコンポーネントは、図 2-4 に示したすべての論理層に含まれるアプリケーションコンポーネントをサポートします。ただし、論理層の概念は、Messaging Server や Calendar Server など、アプリケーションレベルのサービスを提供するシステムサービスコンポーネントにも当てはまります。

論理層について

ここでは、図 2-4 に示した 4 つの論理層について簡単に説明します。この説明では、J2EE™ プラットフォーム (Java 2 Platform, Enterprise Edition) コンポーネントモデルを使用して実装されたコンポーネントを取り上げます。ただし、CORBA など、その他の分散型のコンポーネントモデルも、このアーキテクチャーをサポートしています。

- **クライアント層:** クライアント層は、エンドユーザーがユーザーインターフェースを通して直接アクセスするアプリケーションロジックによって構成されます。クライアント層のロジックには、ブラウザベースのクライアント、デスクトップコンピュータ上で実行される Java コンポーネント、または携帯型のデバイス上で実行される J2ME™ プラットフォーム (Java 2 Platform, Micro Edition) モバイルクライアントが含まれることがあります。
- **プレゼンテーション層:** プレゼンテーション層は、クライアント層に配信するデータを準備し、クライアント層からのバックエンドビジネスロジックへの配信の要求を処理するアプリケーションロジックで構成されます。プレゼンテーション層のロジックは通常、Java サーブレットコンポーネントや JSP コンポーネントなどの J2EE コンポーネントから構成されます。これらのコンポーネントは、HTML 形式や XML 形式の配信用データを準備したり、処理要求を受信したりします。また、この層にはポータルサービスが含まれることもあります。ポータルサービスは、ビジネスサービス層の **ビジネスサービス** へのパーソナル化、セキュリティ保護、およびカスタマイズされたアクセス機能を提供できます。
- **ビジネスサービス層:** ビジネスサービス層は、アプリケーションの主要機能を実行するロジックで構成されます。それらの機能には、データの処理、ビジネスルールの適用、複数ユーザーの調整、データベースや旧バージョンシステムのような外部リソースの管理などがあります。通常、この層は、Java オブジェクト、EJB コンポーネント、メッセージ駆動型 Beans などの J2EE 分散型コンポーネントモデルに準拠する、密接に結合されたコンポーネントで構成されます。個々の J2EE コンポーネントは、在庫情報サービスや税額計算サービスなどの複雑なビジネスサービスを配信するように組み立てることができます。個々のコンポーネントやサービスアセンブリは、サービス指向アーキテクチャーモデル内で緩やかに結合された、SOAP (Simple Object Access Protocol) インタフェース標準に準拠した **Web サービス** としてカプセル化できます。ビジネスサービスは、エンタープライズカレンダーサーバーやメッセージングサーバーのような単独の **サーバー** として構築することもできます。
- **データ層:** データ層は、ビジネスロジックで使用される持続データを提供するサービスで構成されます。このデータは、データベース管理システムに格納されているアプリケーションデータであることも、LDAP (Lightweight Directory Access Protocol) データストアに格納されているリソース情報およびディレクトリ情報であることもあります。このデータサービスには、外部ソースからのデータ供給や旧バージョンのコンピューティングシステムからアクセス可能なデータが含まれることがあります。

論理的および物理的な独立性

図 2-4 に示したアーキテクチャーの次元は、コンポーネントの論理的および物理的な独立性に重点を置いており、4 つの異なる層として表現されています。これらの層は、ネットワーク環境内のさまざまなコンピュータ間でのアプリケーションロジックの区分を表しています。

- 論理的な独立性: アーキテクチャーモデルの 4 つの層は、論理的な独立性を表しています。ある層 (ビジネスサービス層など) のアプリケーションロジックは、ほかの層のロジックとは無関係に変更することができます。プレゼンテーション層またはクライアント層のロジックを変更またはアップグレードしなくても、ビジネスロジックの実行を変更できます。このような独立性により、たとえば、ビジネスサービスコンポーネントを変更しなくても、新しいタイプのクライアントコンポーネントを導入できます。
- 物理的な独立性: 4 つの層は、物理的な独立性も表しています。各層内のロジックは、それぞれ異なるハードウェアプラットフォーム上に (つまり、異なるプロセッサ構成、チップセット、およびオペレーティングシステム上に) 配備できます。この独立性により、個々のコンピュータティング要件および最大限に拡張されたネットワーク帯域幅に最適に対応するように、コンピュータ上の分散型アプリケーションコンポーネントを実行することが可能になります。

アプリケーションコンポーネントやインフラストラクチャーコンポーネントをハードウェア環境 (つまり、配備アーキテクチャー) にマッピングする方法は、ソフトウェアソリューションの規模や複雑さに応じて、さまざまな要因によって決定されます。小規模の配備の場合は、配備アーキテクチャーに含まれるのは数台のコンピュータのみである場合があります。大規模の配備の場合は、ハードウェア環境へのコンポーネントのマッピングには、各コンピュータの速度と演算能力、ネットワークリンクの速度と帯域幅、セキュリティおよびファイアウォールの考慮事項、高可用性およびスケラビリティのためのコンポーネントのレプリケーションの方針などの要素を考慮する場合があります。

システムコンポーネントに適用される層によるアーキテクチャー

図 2-3 で示したように、Java ES インフラストラクチャーサービスコンポーネントは、分散型ソフトウェアソリューションの基盤となるインフラストラクチャーサポートを提供します。ただし、それらのソリューションの一部には、Java ES コンポーネントが直接提供するアプリケーションレベルサービスが含まれます。それらのソリューションは、論理層の設計方法を使用します。

たとえば、Messaging Server が提供する電子メール通信サービスは、いくつかの論理的に区別された Messaging Server 設定を使用して実装されます。これらの異なる設定はそれぞれ、一連の異なるサービスを提供します。メッセージングソリューションを設計するときには、これらの異なる設定は、次の図に示すように別々の論理層に存在する個々のコンポーネントとして表されます。

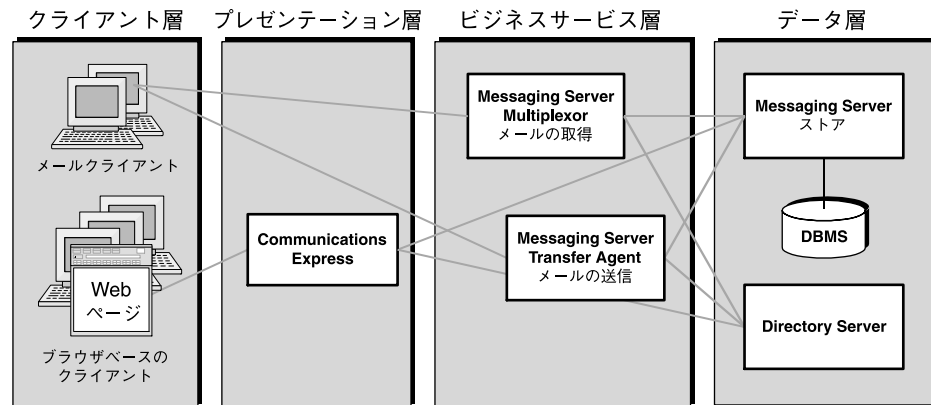


図 2-5 Messaging Server: 層によるアーキテクチャーの例

注 - 図 2-5 は、完全な論理アーキテクチャーを表しているわけではありません。図を簡略化するために、多くの Java ES コンポーネントが省略されています。コンポーネント間を接続する線は、対話を表します。

Messaging Server 機能を論理的に異なる層に分けることにより、Messaging Server の論理的に区別された設定を物理環境内の異なるコンピュータに配備できます。物理的に分離すると、サービス品質の要件 (45 ページの「次元 3: サービスの品質」を参照) を柔軟に満たすことができます。たとえば、インスタンスごとに異なる可用性ソリューションを提供したり、Messaging Server 機能ごとに異なるセキュリティーを実装できます。

次元 3: サービスの品質

すでに説明したアーキテクチャーの 2 つの次元 (インフラストラクチャーサービスの依存関係および論理層) では、主にアーキテクチャーの論理的な面に焦点を当てました。つまり、サービスをエンドユーザーに配信するためにどのような方法で対話するか、あるいはどのようなコンポーネントが必要であるかについてです。一方、配備されるソリューションで同様に重要な次元は、サービス品質の要件を満たすためのソリューションの機能です。

ソリューションアーキテクチャーのサービスの品質の次元は、Java ES サービス品質コンポーネントが果たす役割を明らかにします。

サービス品質

ビジネスの運営におけるインターネットサービスや e コマースサービスの重要性が増しているため、これらのサービスのパフォーマンス、可用性、セキュリティ、スケーラビリティ、および保守性は、高いパフォーマンスを備えた大規模な配備アーキテクチャーの重要なサービス品質要件になりました。

優れたソフトウェアソリューションを設計するには、適切なサービス品質要件を設定し、それらの要件を満たすアーキテクチャーを設計する必要があります。いくつかの重要なサービス品質により、サービス品質要件を指定します。これらのサービス品質を次の表に要約してあります。

表 2-2 ソリューションアーキテクチャーに影響するサービス品質

システムサービス品質	説明
パフォーマンス	ユーザーの負荷条件に関する応答時間の測定
可用性	システムのリソースやサービスがエンドユーザーにアクセス可能になる頻度(システムの「稼働時間」)の測定。
セキュリティ	システムとそのユーザーの整合性を記述する要素の複雑な組み合わせ。セキュリティには、システムの物理セキュリティ、ネットワークセキュリティ、アプリケーションおよびデータのセキュリティ(ユーザーの認証および承認)、またセキュリティ保護された情報のトランスポートも含まれます。
スケーラビリティ	配備されたシステムに対して、随時、容量を拡張する機能。通常、スケーラビリティにはシステムへのリソースの追加が含まれるが、追加時に配備アーキテクチャーを変更する必要はありません。
潜在能力	システムでリソースを追加せずに、異常なピーク負荷使用を処理する機能
保守性	配備されたシステムの保守のしやすさ。システムの監視、発生した問題の修復、ハードウェアおよびソフトウェアのコンポーネントのアップグレードなどが含まれます。

サービス品質の次元は、ソリューションの配備アーキテクチャーに、つまりアプリケーションコンポーネントとインフラストラクチャーコンポーネントが物理環境内でどのように配備されるかに、大きな影響を与えます。

配備アーキテクチャーに影響を与えるサービス品質は、互いに密接に関係しています。あるシステム品質に対する要件がほかのサービス品質の設計に影響を与えることが、よくあります。たとえば、セキュリティのレベルを上げるとパフォーマンスに影響を与える可能性があります。これに伴って可用性にも影響が生じます。冗長性を使用して可用性の問題に対処するためにコンピュータを追加すると、保守コスト(保守性)に影響を与える可能性があります。

ビジネスの要件と制約の両方を満たす配備アーキテクチャーを設計するには、複数のサービス品質が相互に関連する仕組み、およびこれらのかね合いを理解しておくことが重要です。

Java Enterprise System サービス品質コンポーネント

システムサービスコンポーネントまたは分散型アプリケーションコンポーネントが提供するサービス品質を向上するために、主にいくつかの Java ES コンポーネントが使用されます。これらのソフトウェアコンポーネントは、多くの場合、ロードバランサやファイアウォールなどのハードウェアコンポーネントとともに使用されます。

24 ページの「サービス品質コンポーネント」で紹介した Java ES サービス品質コンポーネントについて、次に要約します。

- **可用性コンポーネント:** 可用性コンポーネントは、配備されたソリューションがほぼ連続的に稼働することを可能にします。
- **アクセスコンポーネント:** アクセスコンポーネントは、システムサービスへのインターネットからのセキュリティー保護されたアクセスを可能にし、また多くの場合ルーティング機能も提供します。
- **管理コンポーネント:** 管理コンポーネントは、システムコンポーネントの保守性を向上させます。

次の表は、アーキテクチャーの観点からの最も重要な Java ES サービス品質コンポーネントと、それらの各コンポーネントが最も影響を及ぼすシステム品質を示しています。

表 2-3 サービス品質コンポーネントと影響を受けるシステム品質

コンポーネント	影響を受けるシステム品質
Communications Express	セキュリティー スケーラビリティ
Directory Proxy Server	セキュリティー スケーラビリティ
High Availability Session Store	可用性
Portal Server, Secure Remote Access	セキュリティー スケーラビリティ
Sun Cluster	可用性 スケーラビリティ
Web Proxy Server	セキュリティー パフォーマンス 保守性

Sun Cluster ソフトウェア

Sun Cluster ソフトウェアは、高可用性サービスおよび高スケーラビリティサービスを、Java ES コンポーネントおよび Java ES インフラストラクチャーがサポートするアプリケーションに対して提供します。

クラスタとは緩やかに結合された一連のコンピュータのことであり、サービス、システムリソース、およびデータの単一のクライアントビューを一括して提供します。クラスタの内部では、冗長コンピュータ、インターコネクト、データ記憶域、およびネットワークインタフェースを使用して、クラスタベースのサービスおよびデータに高可用性を提供します。

Sun Cluster ソフトウェアは、メンバーノードおよびその他のクラスタリソースの健全性を継続的に監視します。障害が発生した場合、Sun Cluster ソフトウェアは監視対象のリソースのフェイルオーバーを開始するために介入し、内部の冗長性を使用して、リソースへのほぼ連続的なアクセスを可能にします。

Messaging Server および Calendar Server 用のデータストアサービスをサポートする 2 つのノードから成るクラスタを次の図に示します。

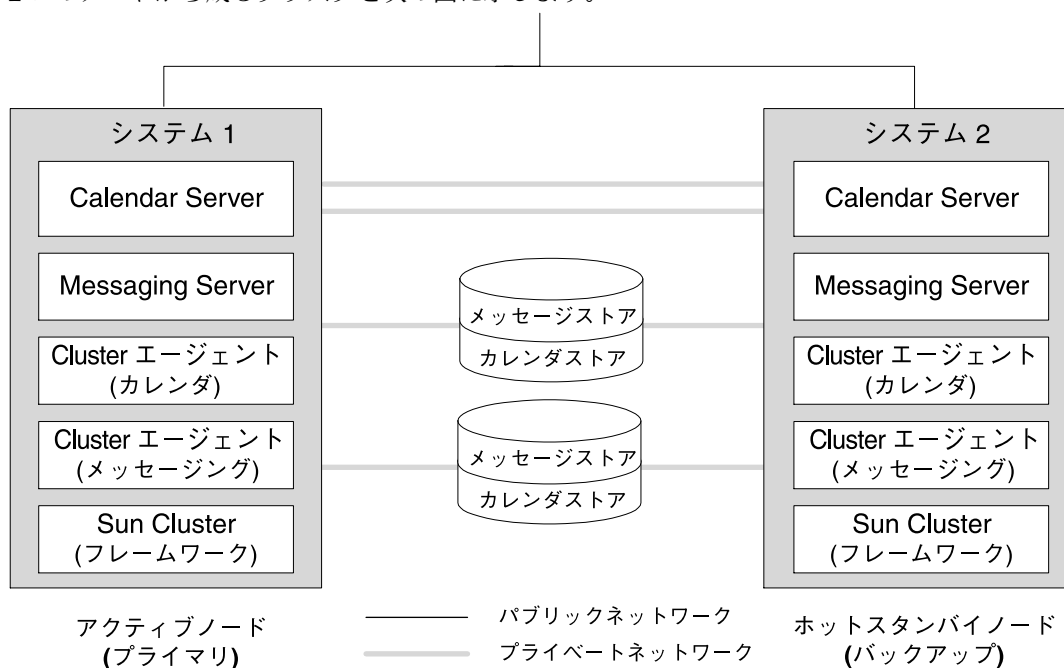


図 2-6 Sun Cluster ノードによる可用性の設計

Sun Cluster データサービスパッケージ (Sun Cluster エージェントと呼ばれることもある) が、すべての Java ES システムサービスコンポーネントに利用できます。カスタム開発されたアプリケーションコンポーネント用のエージェントを記述することもできます。

Sun Cluster ソフトウェアによる制御が行われるので、スケーラブルなサービスも提供できます。クラスタのグローバルファイルシステムおよびクラスタ内の複数のノードの機能を利用して、インフラストラクチャーサービスやアプリケーションサービスを実行することにより、サービスの複数の並行インスタンス間で、これらのサービスに対して増加する要求のバランスを取ることができます。したがって、正しく設定されていれば、Sun Cluster ソフトウェアは分散型のエンタープライズアプリケーションに高可用性とスケーラビリティの両方を提供できます。

Sun Cluster 環境をサポートするのに必要な冗長性のために、ソリューションに Sun Cluster を含めると、物理環境に必要なコンピュータやネットワークリンクの数がかなり増えます。

Sun Cluster の可用性サービスは、ほかの Java ES コンポーネントが提供するサービスとは異なり、分散型のピアツーピアサービスです。したがって、Sun Cluster ソフトウェアは、クラスタ内のすべてのコンピュータにインストールする必要があります。

アーキテクチャーの 3 つの次元の統合

図 2-1 に示し、前のいくつかの節で説明したアーキテクチャーの 3 つの次元を 1 つのものとして捉えると、分散型ソフトウェアソリューションを設計するためのフレームワークが得られます。3 つの次元 (インフラストラクチャーサービスの依存関係、論理層、およびサービス品質) は、ソリューションアーキテクチャーで Java ES コンポーネントが果たす役割を明らかにします。

各次元は、それぞれアーキテクチャーの異なる面を表します。すべてのソリューションアーキテクチャーが、それらのすべての次元を考慮する必要があります。たとえば、ソリューションアーキテクチャーの各論理層の分散型コンポーネント (次元 2) は、適切なインフラストラクチャーコンポーネント (次元 1) と適切なサービス品質コンポーネント (次元 3) のサポートが必要です。

同様に、ソリューションアーキテクチャーに含まれるコンポーネントは、アーキテクチャーの次元ごとに異なる役割を果たします。たとえば、Directory Server はデータ層のバックエンドコンポーネント (次元 2) と持続サービスのプロバイダ (次元 1) の両方とみなされます。

Directory Server はこれらの 2 つの次元の中心に位置するため、この Java ES コンポーネントには、サービスの品質の問題 (次元 3) が最も重要です。Directory Server の障害はビジネスシステムに多大な影響を及ぼすので、このコンポーネントの高可用性設計は非常に重要であり、Directory Server はユーザーや設定に関する機密情報の格納に使用されるため、このコンポーネントのセキュリティの設計も非常に重要です。

Java ES コンポーネントに関するこれらの 3 つの次元の相互作用は、ソリューションの論理アーキテクチャーとソリューションの配備アーキテクチャーの設計に影響します。

35 ページの「Java Enterprise System アーキテクチャーのフレームワーク」で説明したアーキテクチャーフレームワークに基づく詳細な設計方法論を概説することは、このマニュアルの対象外です。ただし、3 次元のアーキテクチャーフレームワークは、Java Enterprise System に基づいたソフトウェアソリューションの配備を理解するのに重要な設計の面を明らかにします。

Java Enterprise System ソリューション アーキテクチャーの例

Java Enterprise System は、広範なソフトウェアソリューションをサポートします。

多くのソリューションは、開発作業を行わずに、Java Enterprise System に含まれるコンポーネントを使ってすぐに設計および配備できます。その他のソリューションには、新しいビジネスまたはプレゼンテーションサービスを提供するカスタム J2EE コンポーネントの開発を必要とする、広範な開発が必要になる場合があります。これらのカスタムコンポーネントを SOAP (Simple Object Access Protocol) インタフェース標準に準拠する Web サービスとしてカプセル化することができます。多くのソリューションは、この 2 つの方法を組み合わせ使用します。

ここでは、前の節のアーキテクチャーの概念に基づき、Java Enterprise System がすぐに使用できるソリューションをどのようにしてサポートするのかを示す例を取り上げます。

エンタープライズ通信のシナリオ

企業は、一般に従業員間の通信、特に電子メールサービスやカレンダーサービスをサポートする必要があります。そのような企業は、内部の Web サイトやその他のリソースに対する個人向けにカスタマイズされたアクセスを企業全体にわたる認証および承認サービスに基づいて従業員に提供すると便利であるとみなします。また、それらの企業は、シングル Web サインオンによりすべてのエンタープライズサービスへのアクセスが可能になるように、すべてのエンタープライズサービスで従業員のアイデンティティを追跡することを望みます。

多数のビジネス要件の一部の例である、このような特定のビジネス要件を次の表にまとめられています。

表 2-4 ビジネス要件の要約: 通信のシナリオ

ビジネス要件	説明	必要な Java ES サービス
シングルサインオン	セキュリティ保護されたエンタープライズリソースおよびサービスへのアクセス (Web アクセスはシングルサインオンの単一の ID に基づく)。	アイデンティティサービス
メッセージング カレンダー	従業員と外部との電子メールメッセージング。 電子的な従業員用のカレンダー機能や会議の設定。	通信サービスと共同作業サービス

表 2-4 ビジネス要件の要約: 通信のシナリオ (続き)

ビジネス要件	説明	必要な Java ES サービス
ポータルアクセス	電子メール、カレンダー、内部 Web ページなどの通信サービスへの単一の Web ベースの個人向けにカスタマイズされたアクセスポイント。	ポータルサービス

さらに、企業には、これらのサービスを提供するソフトウェアシステムのパフォーマンス、可用性、ネットワークセキュリティ、およびスケーラビリティに関する要件があります。

シナリオ例の論理アーキテクチャー

表 2-4 で示したポータルサービス、通信サービス、およびアイデンティティサービスを Java ES コンポーネントを使用して配信するための論理アーキテクチャーを次の図に示します。このアーキテクチャーは、Messaging Server の論理的に異なる設定を、それぞれが異なるサービスを提供するために、別々のコンポーネントとして扱います。

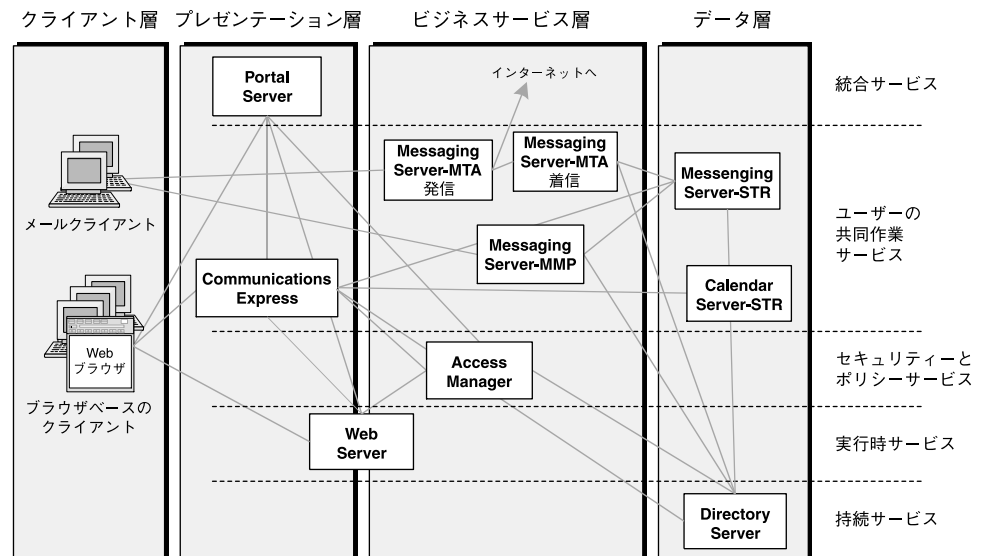


図 2-7 エンタープライズ通信のシナリオの論理アーキテクチャー

コンポーネントは、標準の論理層を表す横の次元、またインフラストラクチャーサービスレベルを表す縦の次元に配置されています。コンポーネント間の対話は、分散型インフラストラクチャーサービスとしての各コンポーネントの機能 (インフラストラクチャーサービスレベル間の対話) または階層アプリケーションアーキテクチャー内の各コンポーネントの役割 (論理層内または論理層間の対話) に依存します。

このアーキテクチャーでは、Directory Server に格納されたユーザー情報にアクセスする Access Manager が、プレゼンテーション層の Portal Server およびその他の Web ベースコンポーネントに対する、シングルサインオン認証および承認の仲介役として機能します。Messaging Server コンポーネントには、データ層のメッセージストア (Messaging Server-STR)、ビジネスサービス層の送信および取得コンポーネント、プレゼンテーション層の HTTP アクセスコンポーネントおよび Communications Express が含まれます。

論理アーキテクチャーは、さまざまな Java ES コンポーネント間のインフラストラクチャーサービスの依存関係も示します。たとえば、Portal Server は、メッセージングチャンネルとカレンダーチャンネルには Communications Express を利用し、認証および承認サービスには Access Manager を利用します。これらのコンポーネントは、今度、ユーザー情報および設定データのために Directory Server を利用します。多数のコンポーネントは、Web Server が提供する Web コンテナサービスを必要とします。

Java ES ソリューションの論理設計の詳細については、『Sun Java Enterprise System 2005Q4 配備計画ガイド』を参照してください。

シナリオ例の配備アーキテクチャー

論理アーキテクチャーから配備アーキテクチャーに移行する際には、サービス品質要件が最も重要になります。たとえば、保護されたサブネットやファイアウォールを使用して、バックエンドデータへのセキュリティーバリアを設けることができます。多くのコンポーネントの可用性とスケーラビリティの要件は、複数のコンピュータにコンポーネントを配備し、ロードバランサを使用してレプリケートしたコンポーネント間に要求を分散することによって満たすことができます。

ただし、より高い可用性要件の水準が適用された場合や大量のディスク容量が必要な場合は、他の可用性ソリューションの方が適しています。たとえば、Messaging Server ストアに Sun Cluster を、Directory Server にマルチマスターレプリケーションを使用できます。

Java ES ソリューションの配備設計の詳細については、『Sun Java Enterprise System 2005Q4 配備計画ガイド』を参照してください。

この章の重要な用語

この節では、この章で使用されている重要な技術用語について説明します。ここでは、用語間の関係や Java Enterprise System の文脈でどのように使用されているかの説明に重点を置いています。

アプリケーションコンポーネント 特定のコンピューティング機能を実行し、ビジネスサービスをエンドユーザーまたはほかのアプリケーションコンポーネントに対して提供する、カスタム開発されたソフトウェアコンポーネント。通常、アプ

リケーションコンポーネントは、CORBA や J2EE™ プラットフォームなどの分散型コンポーネントモデルに準拠しています。これらのコンポーネントを単独であるいは組み合わせて、Web サービスとしてカプセル化できます。

アーキテクチャー	分散型アプリケーション (またはその他のソフトウェアシステム) の論理および物理的な構築ブロックと、それら構築ブロック間の相互関係を示した設計。分散型のエンタープライズアプリケーションの場合、アーキテクチャー設計には一般的に、アプリケーションの論理アーキテクチャーと配備アーキテクチャーの両方が含まれます。
ビジネスサービス	複数のクライアントに代わってビジネスロジックを実行するアプリケーションコンポーネントまたはコンポーネントアセンブリ (したがって、マルチスレッド対応プロセスである)。また、ビジネスサービスは、Web サービスとしてカプセル化された分散型コンポーネントアセンブリであってもかまいませんし、スタンドアロンのサーバーであってもかまいません。
クライアント	ソフトウェアサービスを要求するソフトウェア。(注: ユーザーのことではない。エンドユーザーを参照)。別のサービスを要求するサービス、またはエンドユーザーがアクセスする GUI コンポーネントがクライアントになる場合もあります。
配備アーキテクチャー	論理アーキテクチャーから物理的なコンピューティング環境へのマッピングを示す上位レベルの設計。物理環境には、イントラネット環境またはインターネット環境のコンピュータ、コンピュータ間のネットワークリンク、およびソフトウェアのサポートに必要なその他の物理デバイスが含まれます。
論理アーキテクチャー	分散型アプリケーションの論理的な構築ブロックとそれら構築ブロック間の関係 (またはインタフェース) を記述した設計。論理アーキテクチャーには、分散型アプリケーションコンポーネント、およびこれらのアプリケーションのサポートに必要なインフラストラクチャーサービスの両方が含まれます。
サーバー	分散型のサービスまたは関連する一連のサービスを、外部インタフェースを使ってサービスにアクセスするクライアントに対して提供する、マルチスレッド対応のソフトウェアプロセス。ハードウェアのサーバーとは区別されます。
Web サービス	アクセス可能性、サービスのカプセル化、および検出に関する標準インターネットプロトコルに準拠したサービス。この標準インターネットプロトコルには、SOAP (Simple Object Access Protocol) メッセージングプロトコル、WSDL (Web Service definition Language) インタフェース定義、および UDDI (Universal Discovery, Description, and Integration) レジストリ標準が含まれます。

第 3 章

Java Enterprise System 統合機能

この章では、Java ES コンポーネントを 1 つのソフトウェアシステムに統合する際に主な役割を果たす機能を理解するための概念および技術的な背景について説明します。

これらの機能は、異種インフラストラクチャー製品を手動で統合する場合と比較して、Java Enterprise System を使用するとどのような利点が得られるかを理解するのに役立ちます。

この章では、次の機能を取り上げています。

- 55 ページの「Java Enterprise System 統合インストーラ」
- 57 ページの「統合されたアイデンティティサービスとセキュリティサービス」
- 61 ページの「この章の重要な用語」

Java Enterprise System 統合インストーラ

Java ES のすべてのコンポーネントは、単一のインストーラを使ってインストールされます。このインストーラにより、インストールとアンインストールの手順の一貫性、およびすべてのコンポーネント間の動作の一貫性を保つことができます。

Java ES インストーラは、Java ES ソフトウェアを特定のホストシステムに転送する統合フレームワークです。インストーラを使用すると、コンピューティング環境内のどのコンピュータにも必要な数の Java ES コンポーネントを選択してインストールできます。インストーラは、インストール対象の特定の Java ES コンポーネントに応じて、一定のインストール時の設定も可能にします。

Java ES インストーラそのものは、分散インストールを実行しません。分散型 Java ES ソフトウェアソリューションを配備するには、Java ES インストーラを使用して、環境内のコンピュータごとに1台ずつ、適切なコンポーネントをインストールします。配備アーキテクチャーおよびコンポーネントの依存関係に基づいて、インストールセッションおよび設定手順を適切な順序で使用する必要があります。

このインストーラは、グラフィカルモードおよびテキストベースモードの両方で対話的に実行できる一方、パラメータ駆動型のサイレントインストールモードでも実行できます。このインストーラは英語のほかに、7つの言語をサポートします。フランス語、ドイツ語、スペイン語、韓国語、簡体字中国語、繁体字中国語、および日本語です。

ここでは、Java ES の統合インストーラの次の点について説明します。詳細については、『Sun Java Enterprise System 2005Q4 インストールガイド(UNIX 版)』を参照してください。

- 56 ページの「既存のソフトウェアのチェック」
- 56 ページの「依存性の確認」
- 57 ページの「初期設定」
- 57 ページの「アンインストール」

既存のソフトウェアのチェック

インストーラは、インストール先のコンピュータを検証し、すでにインストールされている Java ES コンポーネントを確認します。次に、インストーラは複数のレベルでチェックを実行して、既存のすべてのコンポーネントが、相互に正常に機能するように適切なリリースレベルにあるかどうかを確認します。これにより、互換性がなく、アップグレードまたは削除する必要があるソフトウェアコンポーネントが表示されません。

同様に、インストーラは、J2SE や NSS など、すでにインストールされている Java ES 共有コンポーネント (26 ページの「共有コンポーネント」を参照) を確認します。インストーラが互換性のないバージョンの共有コンポーネントを検出すると、そのリストが表示されます。インストールを続行すると、インストーラによって、自動的に共有コンポーネントが新しいバージョンにアップグレードされます。

依存性の確認

インストーラは、コンポーネントのチェックを広範囲に行い、ユーザーが選択したインストールコンポーネントが正しく連携して動作するか検証します。

多くのコンポーネントには、ほかのコンポーネントに対する依存性があります。インストーラは、これらの依存性を満たすためのロジックを備えています。このため、インストールするコンポーネントを選択すると、インストーラでは選択したコンポーネントに依存関係のあるコンポーネントおよびサブコンポーネントが自動的に選択されます。

選択したコンポーネント間にローカルな依存関係がある場合、このコンポーネントを選択解除することはできません。ただし、ローカルの依存関係ではない場合は、警告メッセージが表示されますが、別のホストコンピュータ上のコンポーネントによって依存関係が満たされるという前提に基づき、操作は続行できます。

初期設定

多くの Java ES コンポーネントは、初期設定をしてから起動する必要があります。コンポーネントによっては、Java ES インストーラでこの初期設定を実行できます。

インストーラで「今すぐ設定」オプションを選択すると、この初期設定を実行できます。「あとで設定」オプションを選択して初期設定を省略してソフトウェアをインストールすることもできますが、この場合はインストールの完了後に明示的にインストールした各コンポーネントの初期設定を行う必要があります。

インストーラが初期設定を行うようにする場合は、インストール時に必要な設定情報を入力します。具体的には、管理者 ID やパスワードなど、すべてのコンポーネント製品に共通する一連のパラメータ値を指定できます。

アンインストール

Java Enterprise System には、アンインストールプログラムも用意されています。このプログラムを使用して、Java ES のインストーラによってローカルコンピュータにインストールされたコンポーネントを削除できます。アンインストーラはローカルな依存関係を確認し、そのような依存関係が検出された場合は警告メッセージを出力します。アンインストーラは、Java ES 共有コンポーネントを削除しません。

インストーラと同様、アンインストーラはグラフィカルモード、テキストベースモード、またはサイレントモードで実行できます。

統合されたアイデンティティサービスとセキュリティーサービス

Java Enterprise System の重要な機能として、統合されたユーザーアイデンティティ管理と、統合された認証および承認フレームワークが挙げられます。

次の各項では、Java Enterprise System から提供される、統合されたアイデンティティサービスとセキュリティーサービスを理解するための技術的な背景について説明します。

- 58 ページの「単一アイデンティティ」

- 59 ページの「認証とシングルサインオン」

単一アイデンティティ

Java ES 環境では、エンドユーザーに単一統合アイデンティティが割り当てられています。この単一アイデンティティに基づいて、ポータル、Web ページなどの各種リソースへのアクセスや、メッセージング、カレンダー、インスタントメッセージングなどの各種サービスへのアクセスを、ユーザーに許可することができます。

この統合されたアイデンティティおよびセキュリティ機能は、Directory Server、Access Manager、およびその他の Java ES コンポーネント間の緊密な連携動作に基づいています。

Java ES のサービスやリソースへのユーザーアクセスは、ユーザー固有の情報をユーザーリポジトリまたはディレクトリ内の単一のユーザーエントリ内に格納することで実現されます。このユーザー固有の情報には、一意の名前とパスワード、電子メールアドレス、組織内のロール、Web ページの設定などが含まれます。ユーザーエントリ内の情報を使用して、ユーザーを認証したり、特定リソースへのアクセスを承認したり、そのユーザーに各種のサービスを提供したりできます。

Java Enterprise System の場合、Directory Server が提供するディレクトリ内に、ユーザーエントリが格納されます。ユーザーが Java ES コンポーネントから提供されるサービスを要求すると、そのサービスでは Access Manager を使用してユーザーを認証し、特定リソースへのアクセスを承認します。要求されたサービスは、ユーザーのディレクトリエントリ内のユーザー固有の設定情報を確認します。サービスは、その情報を使用してユーザーが要求した作業を実行します。

次の図は、ユーザーの認証および承認を実行するため、またユーザーに対してサービスを提供するためのユーザーエントリへのアクセスを示しています。

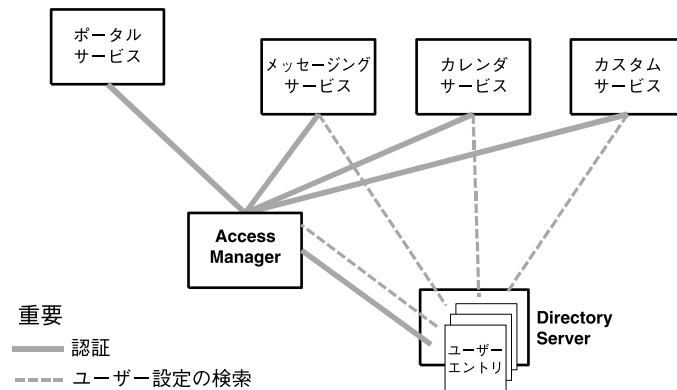


図 3-1 単一ユーザーエントリによる多数のサービスのサポート

このシステムによって可能になる機能の1つに、ユーザーが Web ベースですべての Java ES サービスにサインオンできる機能、つまり他のシステムサービスに対しても自動的に認証される機能があります。シングルサインオンとして知られるこの機能は、Java Enterprise System が提供する強力な機能のうちの1つです。

認証とシングルサインオン

Java ES の認証および承認サービスは、Access Manager によって提供されます。Access Manager は、Directory Server 内の情報を使用することで、ユーザーと Java ES Web サービスまたは企業内のその他の Web ベースのサービスとの対話を仲介します。

Access Manager は、ポリシーエージェントと呼ばれる外部コンポーネントを使用します。ポリシーエージェントは、サービスをホストする Web サーバーまたは Access Manager によってセキュリティ保護されているリソースにプラグインとして追加されます。ユーザーがセキュリティ保護されたリソースに対する要求を送信したときに、ポリシーエージェントは Access Manager の代わりに対話を仲介します。Portal Server や Communications Express といった一部の Java ES コンポーネントでは、Access Manager のサブコンポーネントによってポリシーエージェントの機能が提供されます (74 ページの「[Sun Java System Access Manager 7 2005Q4](#)」を参照)。

認証

Access Manager には、HTTP または HTTPS を介して企業内の Web サービスへのアクセスを要求したユーザーのアイデンティティを検証する認証サービスが含まれています。たとえば、企業の従業員が同僚の電話番号を調べる必要がある場合に、ブラウザを使用してその企業のオンライン電話帳にアクセスします。電話帳サービスにログインするには、ユーザーが自分のユーザー ID とパスワードを入力する必要があります。

認証が行われる順序を図 3-2 に示します。ポリシーエージェントは、電話帳へのログイン要求を仲介し (1)、認証サービスに要求を送信します (2)。認証サービスでは、Directory Server に格納されている情報と照合して、ユーザー ID とパスワードを確認します (3)。ログイン要求が有効と認められると、そのユーザーは認証され (4)、(5)、および (6)、企業の電話帳が従業員に表示されます (7)。ログイン要求が有効と認められない場合は、エラーが生成されて認証に失敗します。

認証サービスは、証明書ベースの HTTPS を介した認証もサポートします。

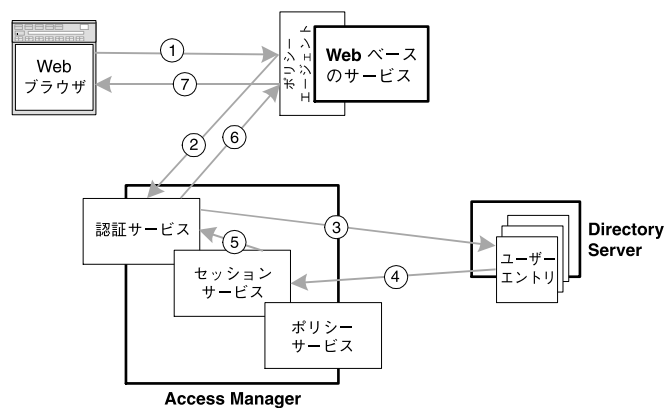


図 3-2 認証順序

シングルサインオン

前の段落で説明した認証の例では、重要な手順を説明しています。ユーザーの認証要求が検証されると、図 3-2 に示すように Access Manager のセッションサービスが起動されます (4)。セッションサービスによってセッショントークンが生成され、ここにユーザーのアイデンティティ情報およびトークン ID が保持されます (5)。セッショントークンはポリシーエージェントに返され (6)、さらに認証要求を発行したブラウザに Cookie として転送されます (7)。

認証されたユーザーがセキュリティ保護された別のサービスにアクセスしようとする時、ブラウザはセッショントークンを対応するポリシーエージェントに渡します。ポリシーエージェントはセッションサービスを使用して、そのユーザーの以前の認証が現在も有効であることを確認します。これにより、ユーザー ID およびパスワードの再入力を求めずに、そのユーザーに対して別のサービスへのアクセスが許可されます。

したがって、ユーザーは 1 回のサインオンだけで、Java Enterprise System が提供する Web ベースの複数のサービスから認証されます。シングルサインオン認証は、ユーザーが明示的にサインオフするか、セッションが期限切れになるまで有効です。

承認

Access Manager には、Java ES 環境の Web ベースのリソースに対するアクセス制御機能を提供するポリシーサービスも含まれています。ポリシーとは、特定の条件下で特定のリソースへのアクセスが承認されるユーザーを記述したルールのことです。次の図に、承認が行なわれる順序を示します。

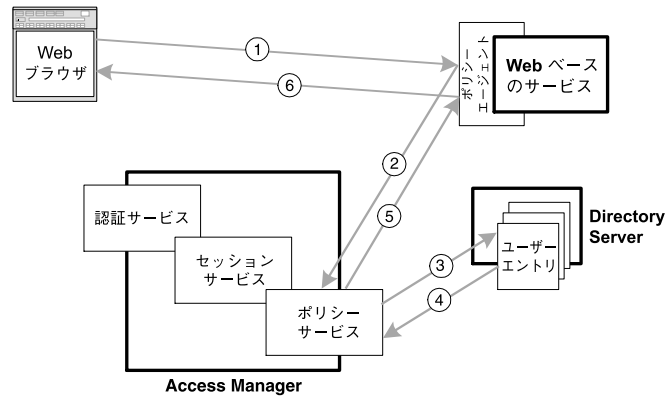


図 3-3 承認順序

認証されたユーザーが Access Manager によってセキュリティー保護されたリソースに対する要求を発行すると (1)、ポリシーエージェントはポリシーサービスに通知します (2)。ポリシーサービスでは、Directory Server の情報を使用して (3)、そのユーザーにリソースへのアクセスポリシーがあるかどうかを確認するために、そのリソースを管理するアクセスポリシーを評価します (4)。ユーザーにアクセス権がある場合は (5)、リソース要求が受け入れられます (6)。

Access Manager を使用すると、1 つの企業内のポリシーを定義、変更、付与、取り消し、および削除することができます。ポリシーは Directory Server に格納され、組織エントリのポリシー関連の属性によって設定されます。ルールは、ユーザーに対して定義したり、ポリシー定義に組み込んだりできます。

ポリシーを適用するのは、Access Manager のポリシーエージェントの役目です。ポリシーサービスがアクセス要求を拒否すると、ポリシーエージェントは要求を発行したユーザーがセキュリティー保護されたリソースへアクセスすることを禁止します。

この章の重要な用語

この節では、この章で使用されている重要な技術用語について説明します。ここでは、用語間の関係や Java Enterprise System の文脈でどのように使用されているかの説明に重点を置いています。

ディレクトリ データの書き込みよりも読み取りに対して最適化された特殊なデータベース。ほとんどのディレクトリは、業界標準のプロトコルである LDAP (Lightweight Directory Access Protocol) に基づいています。

ポリシー	特定の条件下で特定のリソースへのアクセスが承認されるユーザーを記述したルール。このルールは、ユーザーのグループまたは組織内のロールに基づいて決められることもあります。
単一アイデンティティ	Java Enterprise System ディレクトリ内の単一ユーザーエントリに基づいてユーザーが所有するアイデンティティ。ユーザーは、この単一のユーザーエントリに基づいて、ポータルなどの各種の Java Enterprise System リソース、Web ページ、さらにメッセージング、カレンダー、インスタントメッセージングなどのサービスへのアクセスが許可されます。
シングルサインオン	分散型システム内のあるサービスに対するユーザー認証を、システム内のほかのサービスにも自動的に適用できるようにする機能。

第 4 章

Java Enterprise System ソリューションのライフサイクルの作業

この章では、Java ES ソリューションのライフサイクルの各フェーズに関する概念と用語について説明します。この章では、配備作業、特に配備設計作業と配備実装作業に重点を置いています。

この章では、ライフサイクルのフェーズを次の 3 つのグループに分けて説明しています。

- 65 ページの「配備前」
- 66 ページの「配備」
- 71 ページの「配備後」
- 71 ページの「この章の重要な用語」

ソリューションのライフサイクルの作業

ソリューションのライフサイクルについては、Java ES ソフトウェアを使ってビジネスソリューションを実装するための標準的な方法として、第 1 章で紹介しました。この章では、ライフサイクルの各フェーズで行う作業について説明します。簡単に参照できるようにライフサイクルの図を図 4-1 に再度示してあります。

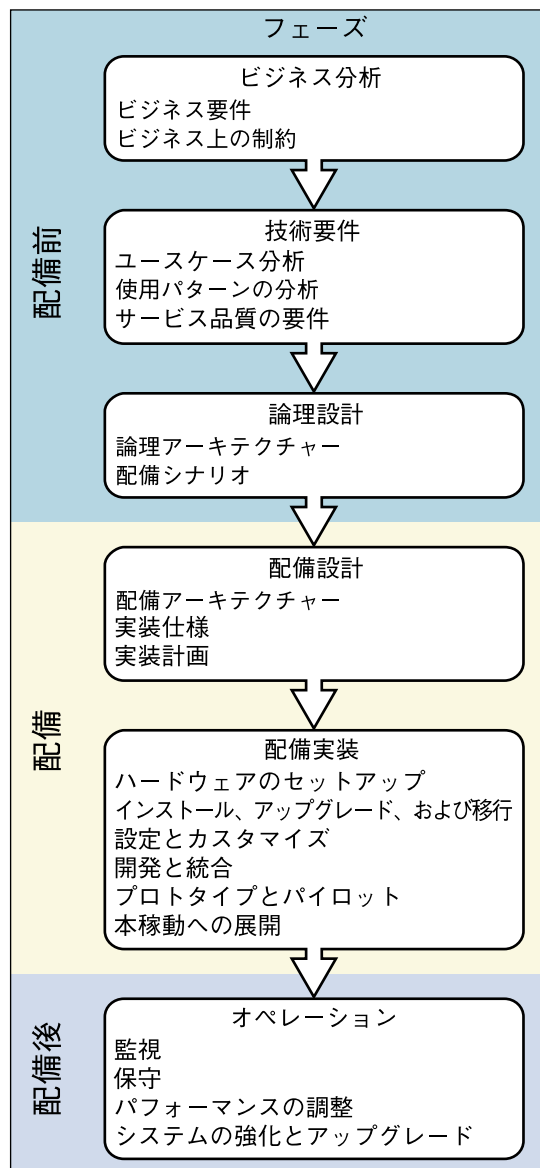


図 4-1 ソリューションのライフサイクルの作業

配備前

ライフサイクルの**配備前**フェーズでは、ビジネスに関するニーズの分析に基づいて**配備シナリオ**を作成します。配備シナリオは、配備を設計するための仕様としての役割を果たします。

配備前の作業は、**図 4-1** に示すように 3 つのフェーズに分けられます。

- **ビジネス分析**: このフェーズでは、提示する配備作業のビジネス上の目的を定義し、その目的を達成するために満たす必要があるビジネス上の要件および制約を記述します。
- **技術要件**: このフェーズでは、ビジネス分析の結果に基づいて**ユースケース**を作成します。ユースケースは、作成予定のソフトウェアシステムとユーザーとの対話をモデル化したものです。これらのユースケースに予測される使用パターンも決定します。ビジネス分析と使用分析の両方に基づいて、提案する配備が満たす必要のあるサービス品質要件 (**表 2-2** を参照) を定式化します。
- **論理設計**: このフェーズでは、技術要件のフェーズで開発したユースケースを分析して、エンドユーザーにサービスを提供するために必要な Java ES インフラストラクチャーコンポーネントおよびカスタム開発されたアプリケーションコンポーネントを特定します。**第 2 章** で説明した概念に基づいて、論理アーキテクチャーを設計します。論理アーキテクチャーは、特定のソフトウェアソリューションのユースケースを実現するために必要なすべてのコンポーネント、およびコンポーネント間のすべての対話を示します。

次の図に示すように、論理アーキテクチャーは、パフォーマンス、可用性、セキュリティ、およびその他のサービス品質要件と組み合わされて、配備シナリオにカプセル化されます。ライフサイクルの配備前のフェーズの詳細については、『Sun Java Enterprise System 2005Q4 配備計画ガイド』を参照してください。

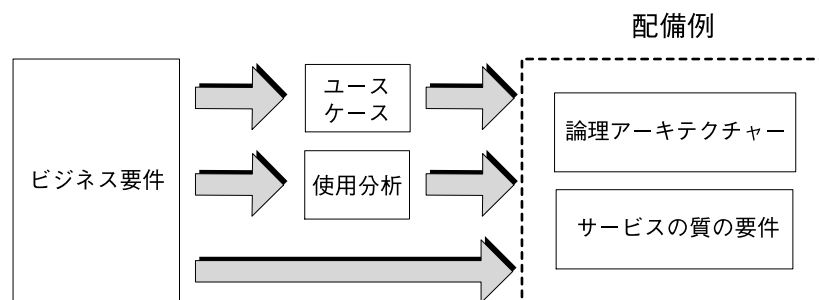


図 4-2 配備シナリオの指定

配備

ライフサイクルの**配備**のフェーズでは、配備シナリオに基づいて配備設計を作成し、その設計の実装、テスト、および本稼働環境への展開を行います。

通常、配備プロセスには、1つのソフトウェアソリューションをサポートするために必要な、すべての層およびすべてのインフラストラクチャーサービスレベルのソフトウェアコンポーネントが関連します。通常、カスタム開発されたアプリケーションコンポーネント (J2EE コンポーネント、Web サービス、またはその他のサーバー) と、ソリューションをサポートするために必要な Java ES コンポーネントの両方を配備する必要があります。

配備の作業は、[図 4-1](#) に示すように 2つのフェーズに分けられます。

- **66 ページの「配備設計」** : 配備設計は、ソリューションの論理アーキテクチャーと、ソリューションが満たす必要があるパフォーマンス、可用性、セキュリティ、スケーラビリティ、保守性、およびその他のサービス品質の要件の両方に依存しています。配備アーキテクチャーにおけるサービス品質の次元は、配備設計フェーズで重要な役割を果たします。
- **68 ページの「配備実装」** : 配備設計の実装は、多くの場合、ハードウェアのセットアップ、ソフトウェアのインストールおよび設定、開発と統合、テスト、および本稼働への展開のその他の作業を含む反復プロセスです。
この後の節で、これらの配備プロセスの2つのフェーズについてさらに詳しく説明します。

配備設計

配備設計フェーズでは、上位レベルの配備アーキテクチャーを作成したあと、下位レベルの実装仕様を作成します。

配備アーキテクチャー

配備アーキテクチャーは、配備シナリオに指定されたサービス品質の要件を満たす方法で、論理アーキテクチャー、つまりアプリケーションの論理的な構築ブロックを物理的なコンピューティング環境にマッピングすることによって作成されます。

次の図に示すように、配備シナリオに基づいて配備アーキテクチャーが作成されます。

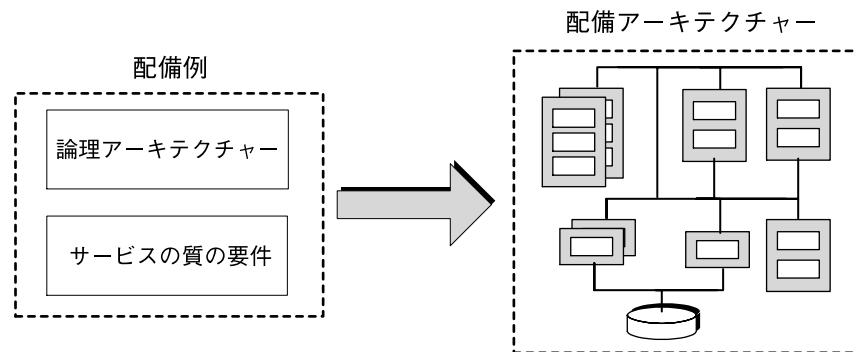


図 4-3 配備シナリオに基づいた配備アーキテクチャの作成

このアーキテクチャ設計の特徴の1つとして、パフォーマンス、可用性、セキュリティ、およびその他のサービス品質の要件を満たすように、物理環境がサイジング(コンピュータの台数を決定し、コンピュータのプロセッサの処理能力およびRAMの要件を見積もる)されます。サイジングが完了した後、物理環境内にあるさまざまなコンピュータにJava ESコンポーネントとアプリケーションコンポーネントを割り当てます。配備アーキテクチャの作成時には、さまざまなコンピュータの能力、システムのインフラストラクチャサービスの特徴、および総所有コストや総利用コストに対する制限を考慮に入れる必要があります。

配備シナリオに含まれるJava ESコンポーネントの数やサービス品質の要件で要求される事項が多ければ多いほど、設計上、より高機能なコンピュータやより広いネットワーク帯域幅が要求されます。ハードウェアが高価でその使用が制限される場合は、固定コスト(ハードウェア)と変動コスト(人的リソースの要件)のバランスやサービス品質の各要件のバランスを再検討したり、設計を見直して効率性を高める必要があります。

配備アーキテクチャの設計では、試行錯誤の繰り返しが必要な場合もあります。ただし、Java Enterprise Systemは配備設計の出発点として、一連の**リファレンス配備アーキテクチャ**を開発します。

参照アーキテクチャは、特定の配備シナリオ、つまり特定のサービス品質の要件が指定された論理アーキテクチャに基づいています。参照アーキテクチャでは、ソフトウェアソリューションは、指定されたサービス品質の要件を満たす方法で、特定の物理環境に配備されます。指定された負荷でのパフォーマンステストは、配備シナリオの開発に使用されたものと同じユースケースのセットに基づいています。参照アーキテクチャのドキュメントは、非開示の条件付きでJava ESの顧客に提供されます。

リファレンス配備アーキテクチャーまたは複数の参照アーキテクチャーの組み合わせに基づいて、ユーザー独自の配備シナリオの要件を満たす配備アーキテクチャーに関する最初の概要を設計できます。ユーザー独自の配備シナリオと参照アーキテクチャーの元になっている配備シナリオの違いを考慮したうえで、参照アーキテクチャーを調整したり、参照アーキテクチャーを参照点として使用したりできます。このようにして、ユーザー独自のサイジングの影響、パフォーマンス、セキュリティ、可用性、容量、および保守性のニーズを評価できます。

実装仕様

実装仕様によって、配備アーキテクチャーの実装に必要な詳細情報が提供されます。この仕様には、通常、次の情報が含まれます。

- コンピュータ、ストレージデバイス、ロードバランサ、ネットワークケーブルなど
の実際のハードウェア
- オペレーティングシステム
- サブネットおよびセキュリティゾーンを含む、ネットワーク設計
- 可用性の設計の詳細
- セキュリティ設計の詳細
- エンドユーザーのプロビジョニングに必要なディレクトリ設計情報

実装計画

実装計画は、配備実装フェーズのさまざまな作業をどのように実行するかを示します。この計画には、通常、次の作業が含まれます。

- ハードウェアのセットアップ
- ソフトウェアのインストール、アップグレード、および移行
- システムの設定とカスタマイズ
- 開発と統合
- テスト
- 本稼動への展開

配備実装

配備設計の実装は、前の節と図 4-1 で示した作業から構成されます。実際の配備プロセスでは作業の反復が発生するので、これらの各作業の順序は固定的なものではありません。以下の各項では、主要な各配備実装作業について、これらが通常実行される順序に従って個別に説明します。これらの作業の詳細については、『Sun Java Enterprise System 2005Q4 ドキュメントロードマップ』を参照してください。

ハードウェアのセットアップ

実装仕様には、物理環境に関するすべての詳細情報が含まれています。たとえば、コンピュータ、ネットワーク設計、ネットワークハードウェア (ケーブル、スイッチ、ルータ、ロードバランサなど)、ストレージデバイスなどです。これらのハードウェアはすべて、Java ES ソリューションをサポートするプラットフォームとして設定する必要があります。

ソフトウェアのインストール、アップグレード、および移行

配備アーキテクチャー、および実装仕様が提供する詳細情報によって、物理環境の各コンピュータに配備するアプリケーションコンポーネントおよび Java ES コンポーネントが明らかになります。Java ES の統合インストーラを使って配備アーキテクチャー内の各コンピュータに適切な Java ES コンポーネントをインストールします (55 ページの「Java Enterprise System 統合インストーラ」を参照)。

インストール計画は、インストーラセッションの順序および範囲を示します。ただし、インストールの実行方法は、初めて Java Enterprise System のインストールを実行するのか、以前インストールした Java ES コンポーネントをアップグレードするのか、またはサードパーティーのコンポーネントを Java Enterprise System と交換するのかどうかによって異なる場合があります。これらの Java ES 導入シナリオの最後の 2 つは、多くの場合、互換性のためにデータやアプリケーションコードの移行を必要とします。

システムの設定とカスタマイズ

さまざまなシステムコンポーネントを統合されたシステムとして連携させるには、いくつかのシステム設定作業を完了させる必要があります。これらの作業の中で最初に完了する必要がある作業は、個々のシステムコンポーネントを起動させるために必要な初期設定です。次に、各 Java ES コンポーネントが対話するコンポーネントと通信できるように設定する必要があります。

各コンポーネントの可用性ソリューションに応じて、高可用性も設定する必要があります。ユーザーが各種サービスにアクセスできるようにユーザーのプロビジョニングを行う必要があります。また、認証と承認のポリシーおよび制御を設定する必要があります (57 ページの「統合されたアイデンティティサービスとセキュリティサービス」を参照)。

ほとんどの場合、設定作業には、要件ごとの機能セットを実現するために、ある程度の Java ES コンポーネントのカスタマイズが必要です。たとえば、通常は、ポータルチャネルを提供するために Portal Server を、承認タスクを実行するために Access Manager を、ウィルスチェックやスパム防止フィルタリングを使用するために Messaging Server を、それぞれカスタマイズします。

開発と統合

一般に、配備シナリオに指定された論理アーキテクチャーによって、ソリューションの実装に必要なカスタム開発作業の範囲が決まります。

配備によっては、Application Server または Web Server 環境で実行される J2EE コンポーネントを使用して新しいビジネスサービスおよびプレゼンテーションサービスを最初から開発する必要があるため、開発は非常に大規模な作業になります。このような場合には、ソリューションのプロトタイプを作成して、開発成果の全体を実装する前に概念実証テストを実行します。

大規模な開発を必要とするソリューション用に、Sun Java Studio では分散型コンポーネントまたはビジネスサービスをプログラミングするツールを提供しています。Sun Java Studio は、Java ES インフラストラクチャーによりサポートされるアプリケーションのプログラミングとテストを簡素化します。

状況によっては、Java ES コンポーネントを旧バージョンのアプリケーションやサードパーティーのサービスと統合できます。このような統合には、データ層の既存のディレクトリまたはデータサービス、あるいはビジネスサービス層の既存のコンポーネントが含まれることがあります。Java ES コンポーネントをこれらのシステムと統合するには、データまたはアプリケーションコードの移行が必要になる場合があります。

J2EE プラットフォームでは、J2EE リソースアダプタを開発することによって既存のアプリケーションを Application Server 環境にプラグインするコネクタフレームワークを提供し、Message Queue では、さまざまなアプリケーションを統合するための強力な非同期メッセージング機能を提供します。

プロトタイプとパイロットのテスト

必要なカスタマイズ作業または開発作業の量によっては、ある時点で配備アーキテクチャーを検証し、ユースケースと比較して、ソリューションがサービス品質要件を満たしているかどうかをテストする必要があります。

カスタム開発サービスが比較的少ない (ほとんど追加設定の必要がない配備) 場合は、ソリューションで単に Java ES コンポーネントのカスタマイズとシステムのパイロットテストのみが必要になることがあります。

一方、重要な新規のアプリケーションロジックを開発してカスタムサービスを作成した場合、このテストはプロトタイプのテスト、統合テストなどを実施する大規模なものになる可能性があります。

このテストによって配備アーキテクチャーの欠点が判明した場合は、アーキテクチャーを修正してテストを再び実行します。最終的に、この反復プロセスによって、本稼動環境への配備が可能な配備アーキテクチャーおよび実装が完成します。

本稼動への展開

本稼動への展開では、本稼動環境に配備実装を組み入れます。このフェーズでは、本稼動環境での分散型アプリケーションおよびインフラストラクチャーサービスのインストール、設定、起動、本稼動システムのエンドユーザーのプロビジョニング、シングルサインオンおよびアクセスポリシーの設定などを行います。通常、まず限定的な配備を行い、その後に組織全体の実装に移行します。このプロセスでは試験稼動を実行し、適用する負荷を徐々に増やして、サービス品質の要件を満たしていることを確認します。

配備後

ライフサイクルの**配備後**の段階では、本稼働環境に配備されたソリューションを実行します。ライフサイクルのオペレーションフェーズでは、次の作業を行います。

- **監視:** システムのパフォーマンスおよびシステムの機能を定期的に監視します。
- **保守:** 新規エンドユーザーのシステムへの追加、パスワードの変更、新規管理ユーザーの追加、アクセス権限の変更、定期的なバックアップの実行などの日常的な管理機能を実行します。
- **パフォーマンスの調整:** システムの運用上のボトルネックを見つけるために定期的に情報を監視し、設定のプロパティの変更、容量の追加などによりボトルネックの除去を試みます。
- **システムの強化とアップグレード:** これらの作業には、システムに新しい Java ES コンポーネントを追加することで新しい機能を追加したり Java ES 以外のコンポーネントを置き換えたりする作業が含まれます。どちらの場合も、これらの変更によりソリューションのライフサイクルの最初のフェーズから、システムの再設計が必要になる場合があります。アップグレード作業は限定的で、通常は Java ES コンポーネントのアップグレードを実行します。

各 Java ES コンポーネントには、それぞれのオペレーションを設定、調整、または管理するための独自の管理ツールが備えられています。監視および管理のための共通のインフラストラクチャーおよびシステム全体を管理するための管理ツールを提供することが目標です。

この章の重要な用語

この節では、この章で使用されている重要な技術用語について説明します。ここでは、用語間の関係や Java Enterprise System の文脈でどのように使用されているかの説明に重点を置いています。

配備	Java Enterprise System ソリューションのライフサイクルプロセスの段階の1つ。この段階では、配備シナリオに基づいて、配備設計の作成、実装、プロトタイプ作成、本稼働環境への展開が行われます。このプロセスの最終的な状態は配備 (配備されたソリューション) とも呼ばれます。
配備シナリオ	特定の Java Enterprise System ソリューションに対する論理アーキテクチャーと、そのソリューションがビジネスニーズを満たすために必要なサービス品質の要件。サービス品質の要件には、パフォーマンス、可用性、セキュリティー、保守性、およびスケーラビリティ/潜在能力に関する要件が含まれます。配備設計は、配備シナリオに基づいて開始されます。
開発	Java Enterprise System ソリューションの配備プロセスに含まれる作業の1つ。この作業によって、配備アーキテクチャーのカスタムコンポーネントがプログラミングおよびテストされます。
配備前	Java Enterprise System ソリューションのライフサイクルプロセスの段階の1つ。この段階では、ビジネスニーズに基づいて配備シナリオが作成されます。このシナリオには、論理アーキテクチャーおよびソリューションが満たす必要のある一連のサービス品質の要件が含まれます。
配備後	Java Enterprise System ソリューションのライフサイクルプロセスの段階の1つ。この段階では、分散型アプリケーションの起動、監視、調整によるパフォーマンスの最適化、および動的アップグレードによる新機能の追加が行われます。
リファレンス配備アーキテクチャー	パフォーマンス確認用に設計、実装、およびテストされた配備アーキテクチャー。リファレンス配備アーキテクチャーは、カスタムソリューションの配備アーキテクチャーを設計するときの開始点として使用されます。
ユースケース	分散型のエンタープライズアプリケーションによって実行される、1つまたは一連のエンドユーザータスク。アプリケーションのパフォーマンスを設計、テスト、および測定するための基礎情報として使用されます。

第 5 章

参照リスト: Java Enterprise System コンポーネント

この付録には、次のカテゴリごとに、すべての Java ES コンポーネントの参照リストが記載されています。

- 73 ページの「システムサービスコンポーネントの説明」: 分散型のエンタープライズアプリケーションをサポートする主な Java ES インフラストラクチャーサービスを提供します。これらのサービスには、19 ページの「Java Enterprise System が必要な理由」で説明したように、ポータルサービス、通信サービスと共同作業サービス、アイデンティティサービスとセキュリティサービス、Web サービスとアプリケーションサービス、および可用性サービスが含まれます。
- 78 ページの「サービス品質コンポーネントの説明」: システムサービスコンポーネントまたは分散型アプリケーションコンポーネントが提供するサービス品質を向上するために使用されます。システムがほぼ連続的に稼動することを可能にする可用性コンポーネント、システムサービスへのセキュリティ保護されたエンドユーザーアクセスを可能にするアクセスコンポーネント、また Java ES ソリューションの保守性の向上に使用するシステム管理コンポーネントがあります。
- 83 ページの「共有コンポーネント」: 特定のホストコンピュータ上で稼動するすべての Java ES コンポーネントが共有できるローカルライブラリです。

この付録では、それぞれのカテゴリおよびサブカテゴリ内で Java ES コンポーネントをアルファベット順に記載しています。

各種コンポーネントのマニュアルに対するロードマップについては、『Sun Java Enterprise System 2005Q4 ドキュメントロードマップ』を参照してください。

システムサービスコンポーネントの説明

Java ES システムサービスコンポーネントは、分散型エンタープライズアプリケーションをサポートするために必要なインフラストラクチャーサービスを提供します。次の各節で Java ES システムサービスコンポーネントについて説明します。

- 74 ページの「Sun Java System Access Manager 7 2005Q4」

- 75 ページの「Sun Java System Application Server Enterprise Edition 8.1 2005Q4」
- 75 ページの「Sun Java System Calendar Server 6 2005Q4」
- 75 ページの「Sun Java System Directory Server 5 2005Q4」
- 76 ページの「Sun Java System Instant Messaging 7 2005Q4」
- 76 ページの「Sun Java System Message Queue 3.6 2005Q4」
- 77 ページの「Sun Java System Messaging Server 6 2005Q4」
- 77 ページの「Sun Java System Portal Server 6 2005Q4」
- 78 ページの「Sun Java System Service Registry 3 2005Q4」
- 78 ページの「Sun Java System Web Server 6.1 2005Q4」

Sun Java System Access Manager 7 2005Q4

Sun Java System Access Manager (Access Manager) は、Web ベースのサービスおよび Web ベースでないアプリケーションを使用する顧客、従業員、およびパートナーのデジタル ID の管理プロセスを、組織が管理するためのインフラストラクチャーを提供します。これらのリソースは内部および外部の広範なコンピューティングネットワークで利用される可能性があるため、ID ごとに属性、ポリシー、資格付与が定義および適用されて、これらのテクノロジーへのアクセスが管理されます。

Java ES インストーラでは、Access Manager はインストール可能な単一のコンポーネントとして提供されます。必要に応じて、次の Access Manager サブコンポーネントは個別にインストールできます。

- **アイデンティティ管理およびポリシーサービスコア:** これを使用すると、ユーザーのアイデンティティを作成および管理したり、ユーザーのアイデンティティに基づいて Java ES リソースへのアクセスを許可するためのポリシーを定義および評価したりできます。このサブコンポーネントには、Access Manager SDK と Delegated Administrator (82 ページの「Sun Java System Delegated Administrator 6 2005Q4」を参照) サブコンポーネントも含まれます。
- **Access Manager SDK:** Access Manager へのリモートインタフェースを提供します。このサブコンポーネントは、Access Manager にリモートアクセスする Java ES コンポーネントをホストするすべてのコンピュータ上にインストールする必要があります。
- **Access Manager 管理コンソール:** アイデンティティサービスとポリシー管理を統合したグラフィカルインタフェースで、ユーザーが Directory Server でユーザーアカウント、サービス属性、アクセス規則を作成、管理するための単一インタフェースとなります。
- **連携管理の共有ドメインサービス:** ユーザーは、複数の関連するサービスプロバイダが提供するアプリケーションに単一の ID でアクセスできます。

Sun Java System Application Server Enterprise Edition 8.1 2005Q4

Sun Java System Application Server (Application Server) は、アプリケーションサービスと Web サービスを開発および配備に使用する J2EE 互換プラットフォームを提供します。Application Server は、リモートメソッドの呼び出しやその他の実行時サービスなど、密接に結合された分散型コンポーネント間で行われる対話用のインフラストラクチャーサービスを提供します。

Java ES インストーラでは、Application Server はインストール可能な単一のコンポーネントとして提供されます。必要に応じて、次の Application Server サブコンポーネントは個別にインストールできます。

- **ドメイン Administration Server:** Application Server の管理や設定、また J2EE コンポーネントおよびアプリケーションの配備などのサーバー側の管理機能を提供します。
- **Application Server 管理クライアント:** Application Server インストールおよびホストされたアプリケーションの管理および設定を可能にする、グラフィカルな管理クライアントを提供します。管理クライアントは、アプリケーションの配備も支援します。
- **コマンド行管理ツール:** Application Server インストールおよびホストされたアプリケーションの管理および設定を可能にする、コマンド行管理クライアントを提供します。このツールは、アプリケーションの配備も支援します。
- **ロードバランスプラグイン。**
- **PointBase:** 持続的なオペレーションに使用できる組み込みデータベースを提供します。
- **アプリケーションのサンプル。**

Sun Java System Calendar Server 6 2005Q4

Sun Java System Calendar Server (Calendar Server) は、企業およびサービスプロバイダ向けの一元化されたカレンダー機能およびスケジュール機能に使用される、スケーラブルな Web ベースのソリューションです。Calendar Server は、個人およびグループのカレンダー機能に加え、会議室や機器などのリソースのカレンダー機能をサポートします。

Java ES インストーラでは、Calendar Server はインストール可能な単一のコンポーネントとして提供されます。

Sun Java System Directory Server 5 2005Q4

Sun Java System Directory Server (Directory Server) は、イントラネット、ネットワーク、およびエクストラネットの情報に対する、一元化されたディレクトリサービスを提供します。Directory Server は既存のシステムに統合され、従業員、顧客、仕入

先、およびパートナー企業の情報を統合化に対応した集中的リポジトリとして機能します。Directory Server を拡張することで、ユーザーのプロファイルや設定情報、およびエクストラネットのユーザー認証を管理できます。

Java ES インストーラでは、Directory Server はインストール可能な単一のコンポーネントとして提供されます。

Sun Java System Instant Messaging 7 2005Q4

Sun Java System Instant Messaging (Instant Messaging) を使用すると、エンドユーザーがインスタントメッセージングやチャットセッションに参加したり、互いにアラートメッセージを送信したり、グループのニュースをすぐに共有したりできます。Instant Messaging は、イントラネットとインターネットの両方に適しており、ほかのインスタントメッセージングプロバイダとの対話をサポートします。

Java ES インストーラでは、Instant Messaging はインストール可能な単一のコンポーネントとして提供されます。次の Instant Messaging サブコンポーネントは個別にインストールできます。

- **Instant Messaging** サーバーコア: サーバーおよびマルチプレクサソフトウェアが含まれます。
- **Instant Messaging** リソース。
- **Access Manager Instant Messaging** サービス。

Sun Java System Message Queue 3.6 2005Q4

Sun Java System Message Queue (Message Queue) は、アプリケーション間通信および信頼性の高いメッセージ配信の問題に対する、標準ベースのソリューションです。Message Queue は、JMS (Java Message Service) オープン標準を実装した企業向けのメッセージングシステムです。

Message Queue の機能は、JMS プロバイダであることに加え、JMS 仕様の最小要件を満たしています。Message Queue ソフトウェアを使用することで、異なるプラットフォームおよびオペレーティングシステム上で稼動するプロセスが共通の Message Queue サービスに接続して、情報を送受信できます。アプリケーション開発者は、ネットワーク間の通信方法に関する低レベルの詳細に注意を奪われることなく、アプリケーションのビジネスロジックに集中して作業を行うことができます。

Message Queue には、次の 2 つのエディションがあります。

- **Enterprise Edition** (デフォルト): マルチプロセッサのメッセージサービス、HTTP および HTTPS 接続、セキュリティ保護されたスケーラブルな接続、クライアント接続のフェイルオーバー、およびクライアントの C 言語対応に対するサポートを提供します。このエディションは、大規模な本稼動環境でのメッセージングアプリケーションの配備および実行に適しています。

- **Platform Edition:** 基本的な JMS サポートを提供し、小規模な配備および開発環境に適しています。
Java ES インストーラでは、Message Queue Enterprise Edition および Message Queue Platform Edition は個別にインストール可能なコンポーネントとして提供されます。

Sun Java System Messaging Server 6 2005Q4

Sun Java System Messaging Server (Messaging Server) は、企業とサービスプロバイダの両方に対応した、強力な標準ベースのインターネットメッセージングサーバーです。Messaging Server は、大容量かつ信頼性の高いメッセージ処理を目的として設計されていて、複数のモジュール (複数の電子メールプロトコルをサポートする個別に設定可能なコンポーネント) から構成されます。

Java ES インストーラでは、Messaging Server はインストール可能な単一のコンポーネントとして提供されます。ただし、インストール後、いくつでも異なるメッセージングサービス、つまり次の Messaging Server サブコンポーネントセットを提供するように各 Messaging Server インスタンスを設定できます。

- **メッセージストア:** メッセージを格納および取得する機能を提供します。
- **Message Transfer Agent (MTA):** SMTP 接続による電子メールの送信、電子メールのルーティング、および適切なメッセージストアへのメッセージの配信をサポートします。電子メールを内部ストレージ (着信) または外部メールストア (発信) に配信するように設定できます。
- **Message Multiplexor (MMP):** IMAP または POP のいずれかのプロトコルを使用して、メッセージストア (またはストアのセット) の電子メールクライアントにアクセスすることにより電子メールの取得をサポートします。
- **Message Express Multiplexor (MEM):** Web ベースの (HTML) 電子メールクライアントによる電子メールの取得および送信をサポートします。

Sun Java System Portal Server 6 2005Q4

Sun Java System Portal Server (Portal Server) は、アイデンティティに対応したポータルサーバーソリューションです。Portal Server はパーソナル化、集約、セキュリティ、統合、検索などの主なポータルサービスを組み合わせます。Portal Server のサブコンポーネントであるモバイルアクセスは、携帯電話や携帯情報端末などのモバイルデバイスから Portal Server へのワイヤレスアクセスを可能にします。

Java ES インストーラでは、モバイルアクセスを含む Portal Server はインストール可能な単一コンポーネントとして提供されます。

Sun Java System Service Registry 3 2005Q4

Sun Java System Service Registry (Service Registry) は、Web サービス (UDDI) レジストリとしても、エンタープライズビジネス XML (ebXML) レジストリとしても機能するリポジトリであり、Web ベースのサービス指向アーキテクチャ (SOA) アプリケーションをサポートします。UDDI レジストリは、Web サービスの登録と検索に使用され、ebXML レジストリは、ビジネスプロセス統合のサポートに必要な情報アーティファクトの格納および管理に使用されます。これらのアーティファクトには、XML スキーマ、ビジネスプロセスルール、Web サービスアクセス制御、バージョン管理、分類スキーマなどのメタデータが含まれます。

Sun Java System Web Server 6.1 2005Q4

Sun Java System Web Server (Web Server) は、マルチプロセスとマルチスレッドに対応するセキュリティ保護された Web サーバーであり、オープン標準に基づいて構築されています。Web Server は、高いパフォーマンス、信頼性、スケーラビリティ、および管理能力を、あらゆる規模の企業に対して提供します。Web Server は、JDK 1.4.1、Java Servlet 2.3、JavaServer Pages™ (JSP™) 1.2、HTTP/1.1、PKCS #11、FIPS-140、168 ビットステップアップ証明書、その他のさまざまなセキュリティベース標準など、広範な Web ソフトウェア標準がサポートします。

Java ES インストーラでは、Web Server はインストール可能な単一のコンポーネントとして提供されます。

サービス品質コンポーネントの説明

Java ES サービスコンポーネントをサポートするコンポーネントは、次のカテゴリに分けられます。

- 可用性コンポーネント
- アクセスコンポーネント
- 管理コンポーネント

これらのカテゴリに含まれるコンポーネントについては、この後の節で説明します。

可用性コンポーネントの説明

可用性コンポーネントは、システムサービスコンポーネントおよびアプリケーションコンポーネントがほぼ連続的に稼動することを可能にします。ここでは、次の Java ES 可用性コンポーネントについて説明します。

- 79 ページの「Sun Cluster 3.1 8/05 と Sun Cluster エージェント」
- 79 ページの「High Availability Session Store 2005Q4」

Sun Cluster 3.1 8/05 と Sun Cluster エージェント

Sun Cluster ソフトウェアは、高可用性サービスおよび高スケーラビリティサービスを、Java Enterprise System と Java ES インフラストラクチャーに基づくアプリケーションに対して提供します。

クラスタとは緩やかに結合された一連のコンピュータ (クラスタノード) のことであり、サービス、システムリソース、およびデータの単一のクライアントビューを一括して提供します。クラスタの内部では、冗長コンピュータ、インターコネクト、データ記憶域、およびネットワークインタフェースを使用して、クラスタベースのサービスおよびデータに高可用性を提供します。Sun Cluster ソフトウェアは、メンバーノードおよびその他のクラスタリソースの健全性を継続的に監視し、障害が発生した場合でも、内部の冗長性を利用してそれらのリソースへのほぼ連続的なアクセスを提供します。

Java ES インストーラでは、Sun Cluster コアおよび Sun Cluster エージェントが、個別にインストール可能なコンポーネントとして提供されます。追加の Sun Cluster エージェントは別の CD に収録されています。

High Availability Session Store 2005Q4

Sun Java System High Availability Session Store (HADB) が提供するデータストアを使用すれば、障害発生時でもアプリケーションのデータが利用可能になります。この機能は、クライアントセッションに関連付けられた状態情報の復元に特に重要です。この機能がないと、セッション中に障害が発生した場合、セッションの再確立時にすべてのオペレーションを繰り返す必要があります。

次の Java ES コンポーネントは、セッション状態情報を格納するサービスを提供します。Application Server、Access Manager、および Message Queue。ただし、これらのコンポーネントの中で、障害時にセッション状態を保持するために HADB サービスを使用できるのは Application Server のみです。

Java ES インストーラでは、HADB はインストール可能な単一のコンポーネントとして提供されます。ただし、HADB サービスを提供するにはサーバーとクライアントの両方のサブコンポーネントが必要です。

アクセスコンポーネントの説明

アクセスコンポーネントは、システムサービスへのフロントエンドアクセスを可能にし、多くの場合、エンタープライズファイアウォールの外にあるインターネットからのアクセスを可能にします。ここでは、次の Java ES アクセスコンポーネントについて説明します。

- 80 ページの「Sun Java System Communications Express 6 2005Q4」
- 80 ページの「Sun Java System Web Proxy Server 4.0.1 2005Q4」
- 80 ページの「Sun Java System Connector for Microsoft Outlook 6 2005Q4」
- 81 ページの「Sun Java System Directory Proxy Server 5 2005Q4」
- 81 ページの「Sun Java System Portal Server, Secure Remote Access 6 2005Q4」

Sun Java System Communications Express 6 2005Q4

Sun Java System Communications Express (Communications Express) は、カレンダーサービス、電子メールサービス、およびアドレス帳サービスへの Web インタフェースをエンドユーザーに提供する、Web ベースの統合された通信および共同作業クライアントを提供します。Communications Express は3つのクライアントモジュールで構成されます。カレンダー、アドレス帳、およびメールです。Communications Express は、Messaging Server または Calendar Server、あるいはその両方にアクセスできるように設定可能であり、Sun Java System LDAP Schema, Version 1 (Schema 1)、Schema 2 のいずれかと連動して動作します。

Java ES インストーラでは、Communications Express はインストール可能な単一のコンポーネントとして提供されます。

Sun Java System Web Proxy Server 4.0.1 2005Q4

Sun Java System Web Proxy Server (Web Proxy Server) は、Web コンテンツのキャッシュ機能、フィルタリング機能、および配信機能を提供します。Web Proxy Server は通常、企業のファイアウォールの内側ではリモートのコンテンツサーバーに対する要求数を低減するために使用され、ファイアウォールの外側では受信されたインターネット要求に対するセキュリティ保護されたゲートウェイを提供するために使用されます。

Java ES インストーラでは、Web Proxy Server はインストール可能な単一のコンポーネントとして提供されます。

Sun Java System Connector for Microsoft Outlook 6 2005Q4

Sun Java System Connector for Microsoft Outlook を使用すると、Outlook を Sun Java Enterprise System のデスクトップクライアントとして使用できます。Connector for Microsoft Outlook は、ユーザーのデスクトップにインストールする必要がある Outlook プラグインです。

Connector for Microsoft Outlook は、Messaging Server に対してフォルダ階層と電子メールメッセージを照会して、この情報を Outlook で表示可能な MAPI (Messaging API) プロパティに変換します。同様に、WCAP を使用して、Calendar Server に対してイベントとタスクを照会し、これらを MAPI プロパティに変換します。Sun Java System Connector for Microsoft Outlook はこのモデルを使って、次の異なる2つの情報ソースからエンドユーザーの Outlook ビューを構築します。Messaging Server のメール情報と Calendar Server のカレンダー情報。

Sun Java System Connector for Microsoft Outlook は、付属の CD に専用のインストーラとともに収録されています。

Sun Java System Directory Proxy Server 5 2005Q4

Sun Java System Directory Proxy Server (Directory Proxy Server) は、e-コマースソリューションに対応したあらゆるミッションクリティカルなディレクトリサービスの中核コンポーネントです。Directory Proxy Server は LDAP アプリケーション層のプロトコルゲートウェイであり、アプリケーション層のロードバランス機能とフェールオーバー機能を利用して、強化されたディレクトリアクセス制御、スキーマ互換性、および高可用性を提供します。

Java ES インストーラでは、Directory Proxy Server はインストール可能な単一のコンポーネントとして提供されます。

Sun Java System Portal Server, Secure Remote Access 6 2005Q4

Sun Java System Portal Server, Secure Remote Access (Portal Server, Secure Remote Access) は、Portal Server のコンテンツとサービスに対するブラウザベースのセキュリティ保護されたリモートアクセスを任意のリモートブラウザに対して提供することで、Portal Server を拡張します。これにより、クライアントソフトウェアを用意する必要がなくなります。Portal Server との統合により、ユーザーはセキュリティ保護されたアクセスで、アクセス権があるコンテンツやサービスにアクセスすることができます。

Java ES インストーラでは、Portal Server, Secure Remote Access はインストール可能な単一のコンポーネントとして提供されます。必要に応じて、次の Portal Server, Secure Remote Access サブコンポーネントは個別にインストールできます。

- **Portal Server, Secure Remote Access** コア。
- **ゲートウェイ:** 外部からのリモートアクセスが可能な企業イントラネットにインタフェースとセキュリティバリアを提供します。ゲートウェイは単一のインタフェースにより、内部 Web サーバーとアプリケーションサーバーのコンテンツをリモートユーザーに安全に提供します。
- **Netlet Proxy:** インターネットなどのセキュリティ保護されていないネットワークを通じて、一般的な TCP/IP サービスを安全に実行できます。Netlet を使用することで、telnet、SMTP、HTTP、固定ポートアプリケーションなどのアプリケーションを実行できます。
- **Rewriter Proxy:** Web リンクを転送し、イントラネットの Web ページの扱いに関するルールセットを作成することで、イントラネット外から企業イントラネットの Web ページへのセキュリティ保護されたアクセスを提供します。

管理コンポーネントの説明

管理コンポーネントは、システムサービスのための、設定や監視などの管理機能を提供します。ここでは、次の Java ES 管理コンポーネントについて説明します。

- 82 ページの「Sun Java System Administration Server (およびコンソール) 5 2005Q4」
- 82 ページの「Sun Java System Directory Preparation Tool 2005Q4」
- 82 ページの「Sun Java System Delegated Administrator 6 2005Q4」

Sun Java System Administration Server (およびコンソール) 5 2005Q4

Sun Java System Administration Server (Administration Server) はサーバーコンソールと連携して、Directory Server、Directory Proxy Server、および Messaging Server を管理するためのグラフィカルツールを提供します。Administration Server は、同一のルートディレクトリ内のサーバーグループにインストールされたサーバーの要求を処理して、要求の実行に必要なプログラムを起動します。

サーバーコンソールは、ネットワーク上の Directory Server インスタンスおよび Administration Server インスタンスと連動して機能するスタンドアロンの Java アプリケーションです。サーバーコンソールは、企業内の Java ES ソフトウェアのフロントエンド管理アプリケーションとして動作します。

Java ES インストーラでは、サーバーコンソールと Administration Server は、まとめてインストール可能な単一のコンポーネントとして提供されます。

Sun Java System Directory Preparation Tool 2005Q4

Sun Java System Directory Preparation Script は、Messaging Server および Calendar Server に対するユーザーのプロビジョニングに必要なスキーマを使って Directory Server を設定するために使用されます。

Sun Java System Delegated Administrator 6 2005Q4

Sun Java System Delegated Administrator は、Calendar Server、Messaging Server、およびその他の Java ES サービスプロバイダに対するユーザー、グループ、ドメイン、およびリソースのプロビジョニングを行うためのコマンド行ユーティリティ (commadmin) です。

Access Manager のインストールを選択すると、Delegated Administrator は自動的にインストールされます。

共有コンポーネント

共有コンポーネントは、Java ES のシステムサービスコンポーネントとサービス品質コンポーネントが依存するローカルサービスとテクノロジサポートを提供します。Java ES インストーラは、ホストコンピュータにインストールされたほかの Java ES コンポーネントをサポートするために必要なすべての共有コンポーネントを自動的にインストールします。

Java Enterprise System には、次の共有コンポーネントが含まれます。

- ANT (Jakarta ANT Java/XML ベースのビルドツール)
- Apache Commons Logging
- Berkeley DB
- 共通のエージェントコンテナ
- ICU (International Components for Unicode)
- J2SE™ (Java 2 Platform, Standard Edition)
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Enterprise Web Application Framework)
- JavaHelp™ Runtime
- JavaMail™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries) Runtime
- JAX-RPC (Java API for XML-based Remote Procedure Call) Runtime
- JCAPI (Java Calendar API)
- JDMK (Java Dynamic Management™ Kit) Runtime
- JSS (Java Security Services)
- KTSE (KT Search Engine)
- LDAP C SDK
- LDAP Java SDK
- MA (Mobile Access) コア
- MFWK (Java ES Monitoring Framework)
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- SAAJ (SOAP with Attachments API for Java)
- SASL (Simple Authentication and Security Layer)
- Sun Explorer Data Collector (Solaris のみ)
- Sun Java Web Console
- WSCL (Web services Common Library)

索引

A

- Access Manager
 - インフラストラクチャーサービスとして, 40
 - システムサービスコンポーネントとして, 23
 - 説明, 74
- Administration Server
 - サービス品質コンポーネントとして, 26
 - 説明, 82
- Ant (Jakarta ANT Java/XML ベースのビルドツール), 83
- Apache SOAP Runtime, 83
- Application Server
 - インフラストラクチャーサービスとして, 40
 - システムサービスコンポーネントとして, 23
 - 説明, 75

B

- Berkeley DB, 83

C

- Calendar Server
 - インフラストラクチャーサービスとして, 40
 - システムサービスコンポーネントとして, 23
 - 説明, 75
- CAPI (Java Calendar API), 83
- Communications Express
 - サービス品質コンポーネントとして, 25
 - 説明, 80

Connector for Microsoft Outlook

- サービス品質コンポーネントとして, 25
- 説明, 80

D

- Delegated Administrator
 - サービス品質コンポーネントとして, 26
 - 説明, 82
- Directory Preparation Tool
 - サービス品質コンポーネントとして, 26
 - 説明, 82
- Directory Proxy Server
 - サービス品質コンポーネントとして, 25
 - システムコンポーネントとして, 47
 - 説明, 81
- Directory Server
 - インフラストラクチャーサービスとして, 40
 - システムサービスコンポーネントとして, 23
 - 説明, 75

E

- EJB コンポーネント, 43

H

- High Availability Session Store
 - サービス品質コンポーネントとして, 25
 - 説明, 79

I

- ICU (International Components for Unicode), 83
- Instant Messaging
 - インフラストラクチャーサービスとして, 40
 - システムサービスコンポーネントとして, 23
 - 説明, 76

J

- J2EE
 - コンポーネント, 43
 - プラットフォーム, 24
 - 分散型コンポーネントモデル, 43
- J2ME プラットフォーム, 43
- J2SE (Java 2 Platform, Standard Edition), 26, 83
- JAF (JavaBeans™ Activation Framework), 83
- JATO (Java Studio Enterprise Web Application Framework), 83
- JavaHelp, 83
- JavaMail, 83
- Java サブレットコンポーネント, 43
- JAX-RPC, 83
- JAXB (Java Architecture for XML Binding), 83
- JAXP (Java API for XML Processing), 83
- JAXR (Java API for XML Registries), 83
- JDMK (Java Dynamic Management™ Kit), 83
- JMS (Java Message Service), 23
- JSP コンポーネント, 43
- JSS (Java Security Services), 26, 83

K

- KT Search Engine (KTSE), 83

L

- LDAP, 43, 61
- LDAP C SDK, 83
- LDAP Java SDK, 83
- Linux, 40

M

- Message Queue
 - インフラストラクチャーサービスとして, 40
 - システムサービスコンポーネントとして, 23
 - 説明, 76
- Messaging Server
 - インフラストラクチャーサービスとして, 40
 - システムサービスコンポーネントとして, 23
 - 説明, 77

N

- NSPR (Netscape Portable Runtime), 26, 83
- NSS (Network Security Services), 26, 83

P

- Portal Server
 - インフラストラクチャーサービスとして, 40
 - システムサービスコンポーネントとして, 24
 - 説明, 77
- Portal Server Secure Remote Access
 - サービス品質コンポーネントとして, 25
 - システムコンポーネントとして, 47
 - 説明, 81

S

- SAAJ (SOAP with Attachments API for Java), 83
- SASL (Simple Authentication and Security Layer), 83
- SOAP (Apache), 83
- Solaris, 40
- Sun Cluster
 - エージェント, 48
 - 可用性サービスとして, 47-49
 - サービス品質コンポーネントとして, 25
 - 説明, 79
- Sun Explorer Data Collector, 83
- Sun Java Web Console, 83
- Sun Java System 製品
 - Access Manager
 - 「Access Manager」を参照

Sun Java System 製品 (続き)

- Administration Server
 - 「Administration Server」を参照
- Application Server
 - 「Application Server」を参照
- Calendar Server
 - 「Calendar Server」を参照
- Communications Express
 - 「Communications Express」を参照
- Connector for Microsoft Outlook
 - 「Connector for Microsoft Outlook」を参照
- Delegated Administrator
 - 「Delegated Administrator」を参照
- Directory Preparation Tool
 - 「Directory Preparation Tool」を参照
- Directory Proxy Server
 - 「Directory Proxy Server」を参照
- Directory Server
 - 「Directory Server」を参照
- High Availability Session Store
 - 「High Availability Session Store」を参照
- Instant Messaging
 - 「Instant Messaging」を参照
- Message Queue
 - 「Message Queue」を参照
- Messaging Server
 - 「Messaging Server」を参照
- Portal Server
 - 「Portal Server」を参照
- Portal Server, Secure Remote Access
 - 「Portal Server, Secure Remote Access」を参照
- Web Server
 - 「Web Server」を参照

W

- Web Server
 - インフラストラクチャーサービスとして, 40
 - システムサービスコンポーネントとして, 24
 - 説明, 78
- Web サービス, 21
 - J2EE コンポーネント、および, 43
 - 定義, 53
- WSCL (Web services Common Library), 83

あ

- アーキテクチャー
 - 概要, 35
 - 次元
 - 「アーキテクチャーの次元」を参照
 - ソリューション, 36
 - 定義, 53
 - 配備, 66-68
 - リファレンス, 67
- アーキテクチャーの次元
 - インフラストラクチャーサービスの依存関係, 37
 - サービスの品質, 45-49
 - 統合, 49
 - 論理層, 42
- アイデンティティ管理, 57
 - サービス, 21, 57-61
 - 単一ユーザー, 58-59
- アクセスコンポーネント
 - 概要, 25-26
 - 説明, 79-81
- アクセスサービス, 21
- アプリケーション
 - エンタープライズ
 - 「分散型のエンタープライズアプリケーション」を参照
 - 分散型
 - 「分散型のエンタープライズアプリケーション」を参照
- アプリケーションコンポーネント
 - 定義, 52
 - 論理層アーキテクチャー内, 42
- アプリケーションサービス, 21, 37
- アンインストーラ, 57

い

- 移行、Java ES 導入シナリオ、および, 31
- 依存関係, 41-42, 57
- 依存性の確認、インストーラ, 56-57
- インストール済みソフトウェアの検出, 56
- インフラストラクチャー
 - サービスの依存関係
 - 「分散型サービス」を参照
 - 分散型のエンタープライズアプリケーション用, 20

え

エンドユーザー

定義, 32

分散型アプリケーション、および, 20

お

オペレーティングシステムサービス, 38

か

開発

定義, 72

配備作業として, 70

可用性

サービス, 47, 79

要件, 46, 47

可用性コンポーネント

概要, 24-25

説明, 78-79

可用性サービス, 21

管理コンポーネント

概要, 26

説明, 82

管理サービス, 21

き

強化, 「導入シナリオ」を参照

共通のエージェントコンテナ, 83

共同作業サービス, 20

共有コンポーネント, 83

概要, 26

定義, 33

く

クライアント

システムサービスコンポーネント、および, 22

定義, 53

クライアント論理層, 43

クラスター

「Sun Cluster」を参照

け

言語サポート, 56

こ

コンポーネント

EJB, 43

J2EE, 43

JSP, 43

アクセス, 25-26

依存関係, 41-42

インストール済みのバージョンの確認, 56

およびインフラストラクチャーサービス, 40

可用性, 24-25

管理, 26

共有, 26, 83

サービス品質, 24-26, 78-82

サブレット, 43

システム

「システムコンポーネント」を参照

システムサービス, 22-24, 73-78

定義, 32

分散型, 19

さ

サーバー

単独の, 43

定義, 53

サービス

Web, 43

インフラストラクチャー, 20

「分散型インフラストラクチャーサービス」を参照

高可用性, 47, 79

スケーラビリティ, 47, 79

定義, 32

サービス品質コンポーネント

概要, 24-26

説明, 78-82

定義, 32

サービス品質要件

可用性, 46, 47

スケーラビリティ, 46, 47

セキュリティ, 46, 47

潜在能力, 46

サービス品質要件 (続き)
パフォーマンス, 46
保守性, 46, 47
作業、Java Enterprise System, 27, 63

し

システム
コンポーネント
「システムコンポーネント」を参照
サービス, 19-21
設定, 57
システムコンポーネント
概要, 22
共有コンポーネント, 26, 83
サービス品質コンポーネント, 24-26
システムサービス, 73-78
システムサービスコンポーネント
「システムサービスコンポーネント」を参照
定義, 33
システムサービス
概要, 22
定義, 33
システムサービスコンポーネント
依存関係, 41-42
概要, 22-24
定義, 33
持続サービス, 38
実行時サービス, 39
実装仕様, 68
承認, 60-61
シングルサインオン
Java ES 機能, 23, 59
インフラストラクチャーサービスレベル, 39
実装, 60
定義, 62

す

スケーラビリティ
サービス, 47, 79
要件, 46, 47

せ

セキュリティ
サービス, 21
ポリシーサービス, 39
要件, 46, 47
潜在能力要件, 46

そ

層、論理
business service, 43
アプリケーションアーキテクチャー、および, 42
クライアント, 43
データ, 43
プレゼンテーション, 43
ソリューション、Java ES
アーキテクチャー, 35
カスタムおよび標準, 50
ライフサイクル, 27-30
例, 50

た

単一アイデンティティ
概要, 58
定義, 62

つ

通信サービス, 20

て

ディレクトリ
定義, 61
ユーザーデータストアとして, 58

と

統合
Java ES 導入シナリオ、および, 31

統合 (続き)

機能

「統合機能」を参照

サービス, 39

統合機能

アイデンティティおよびセキュリティ
, 21, 57-61

概要, 21

共有コンポーネント, 21

統合インストーラ, 21, 55-57

導入シナリオ、Java ES

新しいシステム, 30

アップグレード, 30

概要, 30-31

拡張, 30

強化, 30

定義, 32

トレーニング、Java ES 導入シナリオ、お
よび, 31

に

認証, 59-60

ね

ネットワークトランスポートサービス, 38

は

ハードウェア、Java ES 導入シナリオ、お
よび, 31

配備

アーキテクチャー, 66

開発およびカスタマイズ, 70

実装, 68-71

シナリオ

「配備シナリオ」を参照

設計, 66-68

定義, 72

プロトタイプテスト, 70

本稼働への展開, 71

ライフサイクルの段階, 66-71

ライフサイクルのフェーズ, 66

リファレンスアーキテクチャー, 67

配備アーキテクチャー

概要, 35

設計, 66-68

層によるアーキテクチャーとの関係, 44

定義, 53

配備後

定義, 72

ライフサイクルのフェーズ, 71

配備シナリオ

概要, 65

定義, 72

配備前

定義, 72

ライフサイクルのフェーズ, 65

パフォーマンス要件, 46

ひ

ビジネスサービス

定義, 53

プレゼンテーション層、および, 43

ふ

プラットフォームサービス, 37

プロトタイプ, 70

分散型

アプリケーション

「分散型のエンタープライズアプリケー
ション」を参照

サービス

「分散型サービス」を参照

分散型サービス

Web, 21

アイデンティティ, 21

アクセス, 21

アプリケーションレベル, 37

インフラストラクチャー, 20

概要, 20

可用性, 21

管理, 21

持続, 38

実行時, 21, 39

セキュリティ, 21, 39

通信および共同作業, 20

統合, 39

分散型サービス (続き)
ネットワークトランスポート, 38
プラットフォーム, 37, 38
ポータル, 20
ミドルウェア, 37
メッセージング, 38
ユーザーの共同作業, 39
分散型のエンタープライズアプリケーション
インフラストラクチャー, 20
概要, 19
定義, 32

ほ

ポータルサービス, 20
保守性要件, 46, 47
ポリシー
承認, 60
定義, 62
本稼働への展開, 71

み

ミドルウェアサービス, 37

め

メッセージングサービス, 38

ゆ

ユーザーエントリ, 58
ユーザーカテゴリ
IT マネージャー, 29
システムアナリスト, 29
システムインテグレータ, 29
システム管理者, 29
設計者, 29
代行管理者, 30
特化したシステム管理者, 30
ビジネスプランナ, 29
フィールドエンジニア, 29
ユーザーの共同作業サービス, 39
ユーザーのプロビジョニング, 68

ユーザープロファイル, 29
ユースケース
概要, 65
定義, 72

よ

用語、リンク, 15

ら

ライフサイクルのフェーズ
配備, 29, 66
配備後, 29, 71
配備前, 28, 65

り

リファレンス配備アーキテクチャー
概要, 67
定義, 72

ろ

論理アーキテクチャー
インフラストラクチャーサービスレベル、お
よび, 37
概要, 35
定義, 53
例, 51-52

