



Sun Java™ System

Sun Java Enterprise System 2005Q1

技術の概要

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

Part No: 819-1926

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

Sun Microsystems, Inc. は、この製品に含まれるテクノロジーに関する知的所有権を保持しています。特に限定されることなく、これらの知的所有権は <http://www.sun.com/patents> に記載されている 1 つ以上の米国特許および米国およびその他の国における 1 つ以上の追加特許または特許出願中のものが含まれている場合があります。

このソフトウェアは SUN MICROSYSTEMS, INC. の機密情報と企業秘密を含んでいます。SUN MICROSYSTEMS, INC. の書面による許諾を受けることなく、このソフトウェアを使用、開示、複製することは禁じられています。

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.

この配布には、第三者が開発したソフトウェアが含まれている可能性があります。

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およびその他の国における登録商標です。

Sun、Sun Microsystems、Sun のロゴマーク、Java、Solaris、JDK、Java Naming and Directory Interface、JavaMail、JavaHelp、J2SE、iPlanet、Duke のロゴマーク、Java Coffee Cup のロゴ、Solaris のロゴ、SunTone 認定ロゴマークおよび Sun ONE ロゴマークは、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) の商標もしくは登録商標です。

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。

Legato および Legato のロゴマークは Legato Systems, Inc. の商標であり、Legato NetWorker は同社の商標または登録商標です。

Netscape Communications Corp のロゴマークは Netscape Communications Corporation の商標または登録商標です。

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、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

この製品は、米国の輸出規制に関する法規の適用および管理下にあり、また、米国以外の国の輸出および輸入規制に関する法規の制限を受ける場合があります。核、ミサイル、生物化学兵器もしくは原子力船に関連した使用またはかかる使用者への提供は、直接的にも間接的にも、禁止されています。このソフトウェアを、米国の輸出禁止国へ輸出または再輸出すること、および米国輸出制限対象リスト (輸出が禁止されている個人リスト、特別に指定された国籍者リストを含む) に指定された、法人、または団体に輸出または再輸出することは一切禁止されています。

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。

目次

| | |
|--|-----------|
| 表目次 | 7 |
| 図目次 | 9 |
| はじめに | 11 |
| 対象読者 | 11 |
| マニュアルの使用 | 12 |
| 表記規則 | 14 |
| Web 上の参考資料 | 14 |
| お問い合わせ先 | 15 |
| ご意見、ご要望の送付先 | 15 |
| | |
| 第 1 章 Java Enterprise System の概要 | 17 |
| Java Enterprise System が必要な理由 | 18 |
| Java Enterprise System のコンポーネント | 20 |
| システムサービスコンポーネント | 21 |
| サービス品質コンポーネント | 23 |
| 可用性コンポーネント | 23 |
| アクセスコンポーネント | 23 |
| 管理コンポーネント | 24 |
| 共有コンポーネント | 25 |
| Java Enterprise System での作業 | 25 |
| Java Enterprise System ソリューションのライフサイクル | 26 |
| Java Enterprise System の導入シナリオ | 29 |
| この章の重要な用語 | 31 |
| | |
| 第 2 章 Java Enterprise System ソリューションアーキテクチャ | 33 |

| | |
|--|-----------|
| Java Enterprise System アーキテクチャのフレームワーク | 34 |
| 次元 1: インフラストラクチャサービスの依存関係 | 35 |
| インフラストラクチャサービスレベル | 35 |
| Java Enterprise System インフラストラクチャサービスコンポーネント | 38 |
| Java Enterprise System インフラストラクチャサービスの依存関係 | 39 |
| 次元 2: 論理層 | 41 |
| 論理層について | 42 |
| 論理的および物理的な独立性 | 43 |
| 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 | 43 |
| 次元 3: サービスの品質 | 44 |
| サービス品質 | 45 |
| Java Enterprise System サービス品質コンポーネント | 46 |
| Sun Cluster ソフトウェア | 47 |
| アーキテクチャの 3 つの次元の統合 | 48 |
| Java Enterprise System ソリューションアーキテクチャの例 | 49 |
| 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 | 51 |
| シナリオ例の配備アーキテクチャ | 52 |
| この章の重要な用語 | 53 |
| | |
| 第 3 章 Java Enterprise System 統合機能 | 55 |
| Java Enterprise System の統合インストーラ | 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サービス | 58 |
| 単一アイデンティティ | 58 |
| 認証とシングルサインオン | 59 |
| 認証 | 59 |
| シングルサインオン | 60 |
| 承認 | 61 |
| この章の重要な用語 | 62 |
| | |
| 第 4 章 Java Enterprise System 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 |
| 配備アーキテクチャ | 66 |
| 実装仕様 | 68 |
| 実装計画 | 68 |
| 配備実装 | 69 |

| | |
|--|-----------|
| ハードウェアのセットアップ | 69 |
| 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 | 69 |
| システムの設定とカスタマイズ | 69 |
| 開発と統合 | 70 |
| プロトタイプとパイロットのテスト | 70 |
| 本稼動への展開 | 71 |
| 配備後 | 71 |
| この章の重要な用語 | 72 |
| 付録 A 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 | 74 |
| Sun Java System Access Manager 6 2005Q1 | 74 |
| Sun Java System Application Server Enterprise Edition 8 2005Q1 | 75 |
| Sun Java System Calendar Server 6 2005Q1 | 76 |
| Sun Java System Directory Server 5 2005Q1 | 76 |
| Sun Java System Instant Messaging 7 2005Q1 | 76 |
| Sun Java System Message Queue 3 2005Q1 | 77 |
| Sun Java System Messaging Server 6 2005Q1 | 77 |
| Sun Java System Portal Server 6 2005Q1 | 78 |
| Sun Java System Web Server 6.1 2005Q1 | 78 |
| サービス品質コンポーネントの説明 | 79 |
| 可用性コンポーネントの説明 | 79 |
| Sun Cluster 3.1 9/04 と Sun Cluster エージェント | 79 |
| High Availability Session Store 2005Q1 | 80 |
| アクセスコンポーネントの説明 | 80 |
| Sun Java System Communications Express 2005Q1 | 80 |
| Sun Java System Connector for Microsoft Outlook 6 2005Q1 | 81 |
| Sun Java System Directory Proxy Server 5 2005Q1 | 81 |
| Sun Java System Portal Server, Secure Remote Access 6 2005Q1 | 81 |
| 管理コンポーネントの説明 | 82 |
| Sun Java System Administration Server (およびコンソール) 5 2005Q1 | 82 |
| Sun Java System Directory Preparation Tool 2005Q1 | 83 |
| Sun Java System Delegated Administrator 6 2005Q1 | 83 |
| Sun Remote Services Net Connect 2005Q1 | 83 |
| 共有コンポーネント | 83 |
| 索引 | 85 |

表目次

| | | |
|-------|----------------------------------|----|
| 表 1 | Java Enterprise System のマニュアル | 12 |
| 表 2 | 文字表記の規則 | 14 |
| 表 1-1 | Java ES システムサービスコンポーネント | 21 |
| 表 1-2 | Java ES 可用性コンポーネント | 23 |
| 表 1-3 | Java ES アクセスコンポーネント | 24 |
| 表 1-4 | Java ES 管理コンポーネント | 24 |
| 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 | 28 |
| 表 1-6 | 各 Java ES 導入シナリオに関連する考慮事項 | 30 |
| 表 2-1 | Java ES システムサービスコンポーネント間の関係 | 40 |
| 表 2-2 | ソリューションアーキテクチャに影響するサービス品質 | 45 |
| 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 | 46 |
| 表 2-4 | ビジネス要件の要約 : 通信のシナリオ | 50 |

図目次

| | | |
|-------|-------------------------------------|----|
| 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 | 19 |
| 図 1-2 | Java Enterprise System コンポーネントのカテゴリ | 21 |
| 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 | 27 |
| 図 2-1 | Java ES ソリューションアーキテクチャの次元 | 35 |
| 図 2-2 | 次元 1: インフラストラクチャサービスレベル | 36 |
| 図 2-3 | Java ES システムサービスコンポーネント | 39 |
| 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 | 41 |
| 図 2-5 | Messaging Server: 層によるアーキテクチャの例 | 44 |
| 図 2-6 | Sun Cluster ノードを使用した可用性の設計 | 47 |
| 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 | 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 | 59 |
| 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の作成 | 67 |

はじめに

『Sun Java™ Enterprise System 技術の概要』では、Java Enterprise System の技術および概念上の基盤について説明します。また、Java Enterprise System のコンポーネント、アーキテクチャ、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。

この章で説明する項目は次のとおりです。

- 11 ページの「対象読者」
- 12 ページの「マニュアルの使用」
- 14 ページの「表記規則」
- 14 ページの「Web 上の参考資料」
- 15 ページの「お問い合わせ先」
- 15 ページの「ご意見、ご要望の送付先」

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、または管理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、以下の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

マニュアルの使用

Java Enterprise System のマニュアルは、PDF (Portable Document Format) 形式および HTML (Hypertext Markup Language) 形式のオンラインファイルとして用意されています。どちらの形式のファイルも、障害を持つユーザーにも参照可能です。Sun™ のマニュアルには、次の Web サイトからアクセスできます。

<http://docs.sun.com>

Java Enterprise System のマニュアルには次の場所からアクセスできます。

<http://docs.sun.com/prod/entsys.05q1>

次の表は、Java Enterprise System のマニュアルコレクションに含まれるシステムレベルのマニュアルを示しています。左の列は各マニュアルの書名および Part No. を表す URL、右の列はマニュアルの内容の概要を示しています。

表 1 Java Enterprise System のマニュアル

| マニュアル名 | 内容 |
|--|--|
| 『Java Enterprise System リリースノート』 http://docs.sun.com/doc/819-0815?l=ja | 既知の問題など、Java Enterprise System 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。 |
| 『Java Enterprise System ドキュメントロードマップ』 http://docs.sun.com/doc/819-1912?l=ja | Java Enterprise System に関するマニュアルについて説明しています。ここからコンポーネントに関連するマニュアルにリンクできます。 |
| 『Java Enterprise System 技術の概要』 http://docs.sun.com/doc/819-1926?l=ja | Java Enterprise System の技術および概念上の基盤について説明しています。コンポーネント、アーキテクチャ、プロセス、および機能についても説明しています。 |

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

| マニュアル名 | 内容 |
|---|---|
| 『Java Enterprise System 配備計画ガイド』 http://docs.sun.com/doc/819-1919?l=ja | Java Enterprise System 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 Enterprise System 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。 |
| 『Java Enterprise System User Management Guide』 http://docs.sun.com/doc/817-5761 | Java Enterprise System ソリューションのユーザーについての情報の計画、配備、および管理に役立ちます。ソリューションのライフサイクルの各フェーズでのユーザー管理の問題について説明し、『Java Enterprise System 配備計画ガイド』を補います。 |
| 『Java Enterprise System 配備実例集 : 評価のシナリオ』 http://docs.sun.com/doc/819-1666?l=ja | Java Enterprise System 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。 |
| 『Java Enterprise System インストールガイド』 http://docs.sun.com/doc/819-0808?l=ja | Java Enterprise System 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。インストールするコンポーネントの選択方法、インストール後のコンポーネントの設定方法、また設定したコンポーネントが適切に動作しているかどうかの確認方法を示しています。 |
| 『Java Enterprise System アップグレードと移行』 http://docs.sun.com/doc/819-2235?l=ja | 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の Java Enterprise System をアップグレードするための情報および手順を示しています。 |
| 『Java Enterprise System 用語集』 http://docs.sun.com/doc/819-1933?l=ja | Java Enterprise System のマニュアルで使用される用語を定義しています。 |

この表に記載されているシステムレベルのマニュアルの他に、Java Enterprise System のマニュアルセットには各 Java Enterprise System コンポーネント用のコンポーネント固有のマニュアルも含まれています。詳細は『Java Enterprise System ドキュメントロードマップ』を参照してください。

表記規則

次の表は、このマニュアルで使用される文字表記の規則を示しています。

表 2 文字表記の規則

| 表記 | 意味 | 例 |
|-----------------------------|---|--|
| AaBbCc123 (モノスペース) | API および言語の要素、HTML タグ、Web サイトの URL、コマンド名、ファイル名、ディレクトリパス名、画面出力の表示、サンプルコード | .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。 % You have mail. |
| <i>AaBbCc123</i> (イタリック) | 実際の名前や値に置き換えられるコマンド行変数 | ファイルは、 <i>install-dir/bin</i> ディレクトリに格納されています。 |

Web 上の参考資料

次の場所には、Java Enterprise System およびそのコンポーネント製品に関する情報が用意されています。

<http://www.sun.com/software/javaenterprisesystem/index.html>

お問い合わせ先

Java Enterprise System の使用にあたって問題が発生した場合は、次のいずれかの方法で Sun のカスタマーサポートに連絡してください。

- Sun Software Support サービス

<http://www.sun.com/service/sunone/software>

このサイトには、「Knowledge Base」、「Online Support Center」、および「ProductTracker」へのリンクがあります。また、保守プログラム、およびサポートの問い合わせ先電話番号を参照することもできます。

- 保守契約に規定されている緊急電話番号

できるだけ適切に問題に対処するために、お問い合わせの際には次の情報をお知らせください。

- 問題の詳細な内容 (発生状況や業務への影響の度合など)
- 使用しているマシンの種類、オペレーティングシステムのバージョン、製品のバージョン (すべてのパッチや、問題に影響している可能性のあるその他のソフトウェアを含む)
- 問題を再現するための詳細な手順
- エラーログまたはコアダンプ

ご意見、ご要望の送付先

Sun ではマニュアルの品質向上のため、お客様のご意見、ご要望をお受けしております。

ご意見は、<http://docs.sun.com> にアクセスして、「コメントの送信」をクリックしてお送りください。オンラインフォームにマニュアルのタイトルと Part No. を記入してください。Part No. は、マニュアルのタイトルページまたは最上部に記載されている 7 桁または 9 桁の番号です。たとえば、このマニュアルのタイトルは『Sun Java Enterprise System 技術の概要』であり、Part No. は 819-1926 です。

ご意見、ご要望の送付先

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。

- 18 ページの「Java Enterprise System が必要な理由」
- 20 ページの「Java Enterprise System のコンポーネント」
- 25 ページの「Java Enterprise System での作業」
- 31 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。

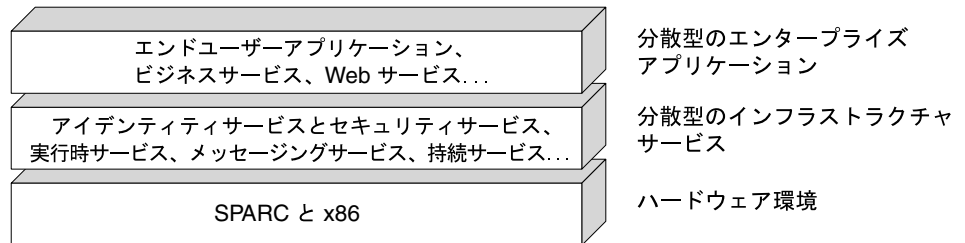
Java Enterprise System では、このような**分散型のエンタープライズアプリケーション**、つまり、通常次のような特徴を持つアプリケーションをサポートするインフラストラクチャサービスを提供します。

- **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を介して配備された対話型ソフトウェア**コンポーネント**で構成されています。これらの分散型コンポーネントは、環境内のさまざまなコンピュータで実行されます。分散型の各コンポーネントが相互に連携して、**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向けの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スケールが、本稼動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には、基本となる一連のインフラストラクチャ**サービス**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できます。さらに、これらのインフラストラクチャサービスは、コンピュータおよびネットワークリンクのハードウェア環境によってサポートされています。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。ほとんどの場合、Java Enterprise System は、[図 1-1](#) に示す分散型インフラストラクチャ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：** ポータルサービスを利用することで、従業員、在宅勤務者、知識労働者、パートナー企業、仕入先、および顧客が、企業ネットワークの内外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：** 多様なユーザーコミュニティ間で情報を安全に交換できるようにします。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ネットワーク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、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：** 分散型コンポーネントが実行時に互いにやり取りできるようにし、広範な種類のサーバー、クライアント、およびデバイス用のアプリケーションを開発、配備、管理できるようにします。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：** アプリケーションコンポーネント、またアプリケーションコンポーネントをサポートするインフラストラクチャ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。

- **アクセスサービス：** 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可能にします。
- **管理サービス：** 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の保守および調整に役立ちます。

各サービスがいくつかの Java Enterprise System コンポーネントから構成される、1つ以上の Java Enterprise System サービスを配備できます。

Java Enterprise System のコンポーネント

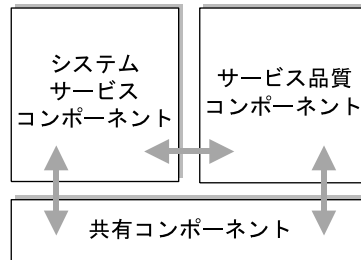
Java Enterprise System は、以前は個別に提供されていたソフトウェア製品を1つのソフトウェアシステムに統合したものです。このシステムのコンポーネントに対するテストが実施されていて、各コンポーネント間の相互動作が保証されています。次のような多数のシステムレベルの機能によって、これらの統合が可能になりました。

- 共有ライブラリの共通セットですべてのコンポーネントの同期をとります。
- Java Enterprise System のすべてのコンポーネントは、1つのインストーラを使用してインストールされます。
- すべての Java Enterprise System コンポーネントが、統合されたユーザーアイデンティティとセキュリティ管理システムを共有できます。

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きます。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つの主要なカテゴリに分けることができます。

- **システムサービスコンポーネント：** 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主要な Java Enterprise System インフラストラクチャ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、およびその他のサービス品質を向上させます。
- **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 Enterprise System 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 Enterprise System 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。**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、J2EE アプリケーションサービスがあります。

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**、およびシステムサービスコンポーネントが提供するサービスについての簡単な説明が次の表にあります。各システムサービスコンポーネントは、多数のクライアントをサポートできる、マルチスレッド対応のサーバープロセスです。各種のコンポーネントの詳細については、[74 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、およびアプリケーションやサービスへのアクセスに対するロールに基づいた認証が含まれます。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 も、Web コンテナサービスを提供します。 |

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 Calendar Server | エンドユーザーおよびエンドユーザーのグループにカレンダーサービスおよびスケジューリングサービスを提供します。 Calendar Server には、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ています。 |
| Sun Java System Directory Server | アイデンティティプロファイル (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、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 | 緩やかに結合した分散コンポーネントと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、セキュリティ、スケーラビリティ、遠隔管理などのエンタープライズ機能を追加します。 |
| Sun Java System Messaging Server | 電子メール、FAX、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、複数のメッセージ記憶域への並行アクセスが可能であり、一方的に送られてくる電子メールの受信拒否やウイルスの攻撃の予防に役立つコンテンツフィルタリングを提供します。 |
| Sun Java System Portal Server | ブラウザからビジネスアプリケーションやサービスにアクセスするクライアントに対して、コンテンツの集約や個人用のカスタマイズなどの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 サー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 コンポーネント用の 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 Page など、スタティックおよびダイナミックな 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。 |

サービス品質コンポーネント

表 1-1 に示したシステムサービスコンポーネントに加えて、Java Enterprise System にはシステムサービスコンポーネントまたはカスタム開発アプリケーションコンポーネントが提供するサービスの品質を向上させるために使用するコンポーネントがいくつか含まれています。これらのサービス品質コンポーネントは、次のように分類されます。

- 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。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。各種のコンポーネントの詳細については、79 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。各種のコンポーネントの詳細については、80 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 Directory Server 向けのセキュリティサービスを提供します。Directory Proxy Server は、強化されたディレクトリアクセス制御、スキーマ互換性、属性のフィルタリング、および複数の Directory Server インスタンスのルーティングを提供します。 |
| Sun Java System Portal Server, Secure Remote Access | 企業ファイアウォールの外から、内部ポータルやインターネットアプリケーションを含む Portal Server のコンテンツおよびサービスへの、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 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可能にします。 |
| 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 および Calendar Server の両方への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。各種の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 | ユーザーを Messaging Server および Calendar Server に対してプロビジョニングするために必要なスキーマを Directory Server に設定する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に入力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。 |

表 1-4 Java ES 管理コンポーネント (続き)

| コンポーネント | 提供される管理サービス |
|---------------------------------|-----------------|
| Sun Remote Services Net Connect | リモート監視機能を提供します。 |

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たいくつかの共有ライブラリが含まれます。Java Enterprise System **共有コンポーネント**は、同じホストコンピュータ上で稼動する Java Enterprise System 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 の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。完全なリストについては、[83 ページの「共有コンポーネント」](#)を参照してください。

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。これらの作業の範囲および難易度は、Java Enterprise System の導入時の出発点、また作成し配備するソリューションの性質により異なります。

ここでは、Java Enterprise System の 2 つの面、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 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作業は、[27 ページの図 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](#) 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を作成します。このアーキテクチャ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は、ソフトウェアソリューションを本稼動環境に配備（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動状態で実行し、パフォーマンスを監視および最適化します。必要な場合はアップグレードを実行し、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「Java Enterprise System ソリューションのライフサイクルの作業」](#) で詳しく説明します。

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

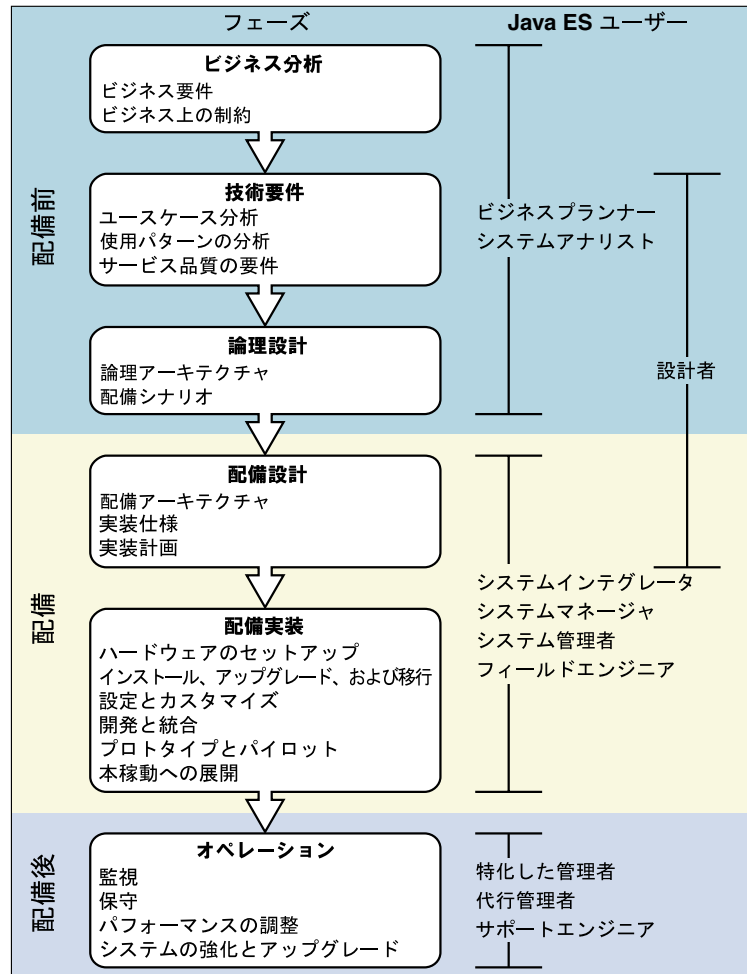


図 1-3 は、ライフサイクルの各フェーズの作業を通常実行する Java Enterprise System ユーザーを示しています。Java Enterprise System で作業を実行する場合、実行するジョブは図 1-3 に示す 1 つ以上のユーザーカテゴリに該当します。各カテゴリのユーザーのスキルとバックグラウンドは、次の表のとおりです。

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 |
|-----------------------------------|--|------------------------|
| ビジネスプランナー 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 | 専門的である。 IT 環境に精通している。 | 配備設計 配備実装 |
| システム管理者 システムマネージャ | 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、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。 | |
| 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 | 特化した技術知識または製品知識を持つ。 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。 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。 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。 | オペレーション |

Java Enterprise System の導入シナリオ

Java Enterprise System 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 Enterprise System 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 IT インフラストラクチャで始め、そのシステムの一部、多くの部分、または全部を Java Enterprise System 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で始め、現在の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 Enterprise System 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：** 新しいソフトウェアで既存のインフラストラクチャを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、ソフトウェアサブシステムを置き換えるには、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と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：** ほとんどのインフラストラクチャの変更は、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 テクノロジー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るか強化するときには、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場合があります。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 Enterprise System 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 各 Java ES 導入シナリオに関連する考慮事項

| 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* |
| 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ります | 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 |

* 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。

エンドユーザー：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イスを通して、分散型アプリケーションを使用するユーザーのこと。アプリケーションが何人のエンドユーザーを同時にサポートするかは、そのアプリケーションの**配備アーキテクチャ**を決定する上で重要な要因となります。

共有コンポーネント：Java Enterprise System に含まれる**システムコンポーネント**の1種。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。一方、**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サービスを提供します。

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**、または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。それらのコンポーネントを単独であるいは組み合わせ、**ビジネスサービス**を提供したり、**Web サービス**としてカプセル化できます。

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ー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**システムコンポーネント**の1種。アクセスコンポーネントと管理コンポーネントを含む、サポートコンポーネントは、**システムサービスコンポーネント**のサポートを行います。

システムコンポーネント：Java Enterprise System に格納されていて、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。分散型インフラストラクチャ**サービス**を提供する**システムサービスコンポーネント**、アクセスサービスおよび管理サービスを提供することによってシステムサービスをサポートする**サービス品質コンポーネント**、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、さまざまな種類のシステムコンポーネントが存在します。

システムサービス：Java Enterprise System から提供される固有の機能を定義する1つまたは複数の分散型**サービス**。通常、システムサービスを実行するには、多数の**サービス品質コンポーネント**、多数の**共有コンポーネント**、またはその両方からサポートされている必要があります。

システムサービスコンポーネント : Java Enterprise System に含まれる **システムコンポーネント** の 1 種。システムサービスコンポーネントは、主な Java Enterprise System インフラストラクチャサービスを提供します。それらのサービスには、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 サービスとアプリケーションサービス、可用性サービスが含まれます。

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であり、着手するソフトウェアシステムや達成する目標を明らかにします。Java Enterprise System には、新しいシステム、置き換え、拡張、およびアップグレードの 4 つの基本的な導入シナリオがあります。

分散型のエンタープライズアプリケーション : 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を対象にしている (分散型)、適用範囲とスケールが本稼動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 (エンタープライズ向け) アプリケーション。

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

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 ソリューションのアーキテクチャには、論理アーキテクチャと配備アーキテクチャの 2 つの面があります。論理アーキテクチャは、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は、論理アーキテクチャの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と配備アーキテクチャ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の設計のためのアーキテクチャ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。

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

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

Java Enterprise System 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。

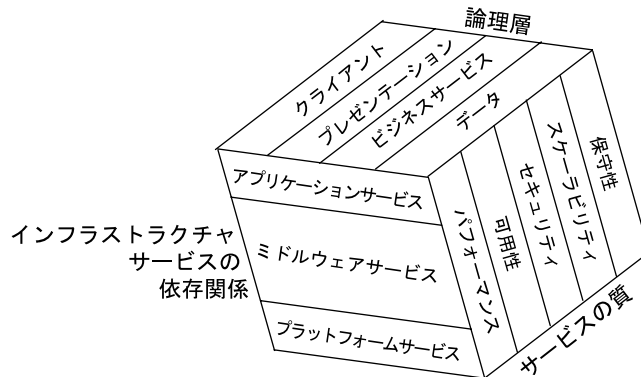
ビジネス要件によって課されるパフォーマンス、可用性、セキュリティ、スケーラビリティ、および保守性のレベルに必要な機能を実現するには、これらのソフトウェアソリューションを適切に設計する必要があります。

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で表します。特に、分散型システムの設計にはアーキテクチャの次の3つの次元が関係します。

- **インフラストラクチャ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をサポートするシステムサービスコンポーネント ([21 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、可用性、セキュリティ、スケーラビリティ、保守性などのサービスの品質の要件がどのように実現されるかについて、サービス品質コンポーネントの役割 ([23 ページの「サービス品質コンポーネント」](#)を参照) を含めて、重点を置いています。

ソリューションアーキテクチャのこれらの3つの次元を次の図に示します。

図 2-1 Java ES ソリューションアーキテクチャの次元



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サービスレベル

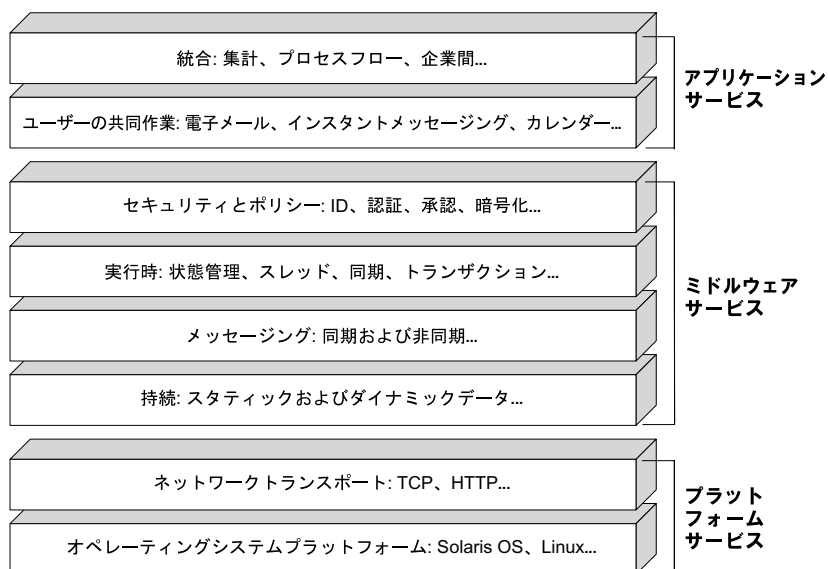
分散型ソフトウェアシステムを設計する場合、ほとんどカスタム開発コンポーネントで構成されるか、または追加設定の必要ないコンポーネントで構成されるかに関わらず、いくつかのインフラストラクチャサービスを統合する必要があります。これらのサービスは、多数のレベルで機能します。

ソリューションアーキテクチャのインフラストラクチャサービスの依存関係の次元を [36 ページの図 2-2](#) に示します。この図に示されているレベルは、[19 ページの図 1-1](#) のインフラストラクチャサービス層の詳細を示しています。

図 2-2 中のサービスの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サービスは、Java ES システムサービスコンポーネント (21 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盤になります。

一般的に、図 2-2 に示したサービスは、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、および他の 2 つのグループの間にあることから名前が付けられた、ミドルウェアサービスのグループという、3 つのおおまかなグループに分けられます。

図 2-2 次元 1: インフラストラクチャサービスレベル



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 の最下位レベルのサービス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する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、物理デバイスの他に、メモリ、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る分散型のアプリケーションコンポーネント間の通信に対する、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せん。
- **持続：**スタティックデータ（ユーザー、ディレクトリ、設定などの情報）およびダイナミックアプリケーションデータ（頻繁に更新される情報）へのアクセス、およびその格納に対する基本的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保護されたアクセスをサポートします。これらのサービスには、**シングルサインオン**機能の他に、分散しているリソースへのグループベースまたはロールベースのアクセスを管理するポリシーに対するサポートも含まれます。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。

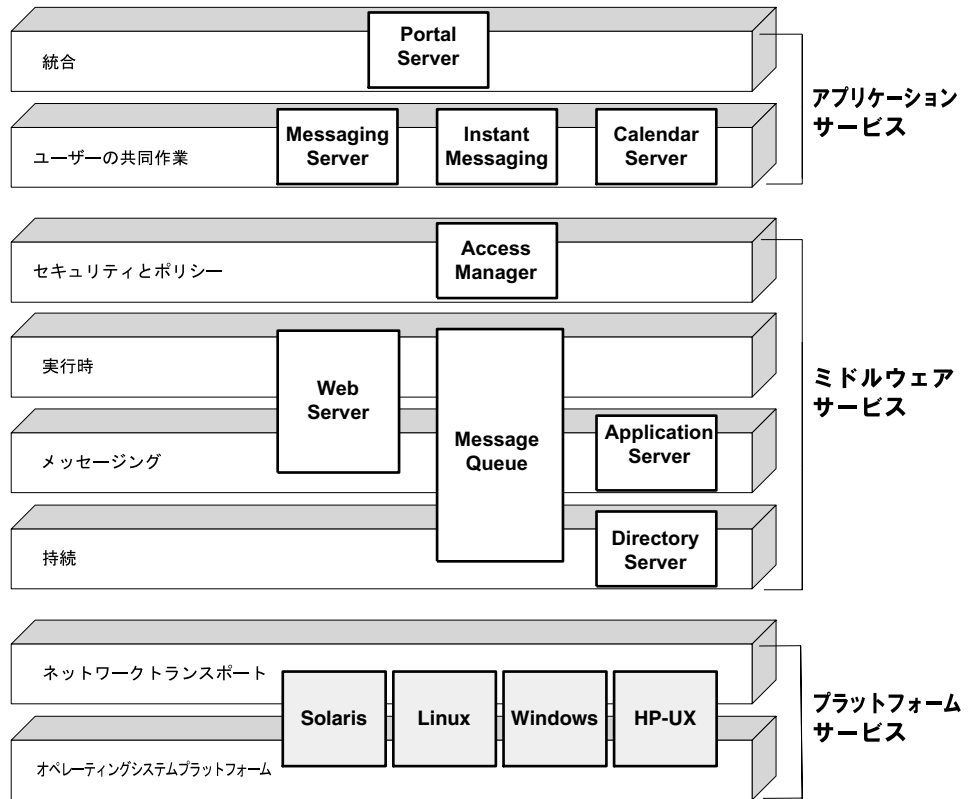
図 2-2 に示したサービスレベルは、最下位レベルのオペレーティングシステムサービスから最上位レベルのアプリケーションサービスや統合サービスまでの、さまざまなインフラストラクチャサービス間の一般的な依存関係を反映しています。通常、各サービスはその下にあるサービスに依存し、その上にあるサービスをサポートします。

ただし、図 2-2 は、インフラストラクチャサービスの層を厳密に表していません。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サービスコンポーネント

Java ES コンポーネントによって、図 2-2 に示す分散型インフラストラクチャサービスレベルが実装されます。さまざまなレベルにおける Java ES システムサービスコンポーネントの位置付けを図 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 に示す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 コンポーネントがサポートされ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含まれています。

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

一般的に、図 2-3 に示す各 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、最上位のシステムサービスコンポーネント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 対応するチャン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ンネル用) |
| 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ンネル用) |
| 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 (カレンダーチャンネル用) |
| 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 |
| Web Server | Access Manager (アクセス制御用) | Portal Server Access Manager |

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)

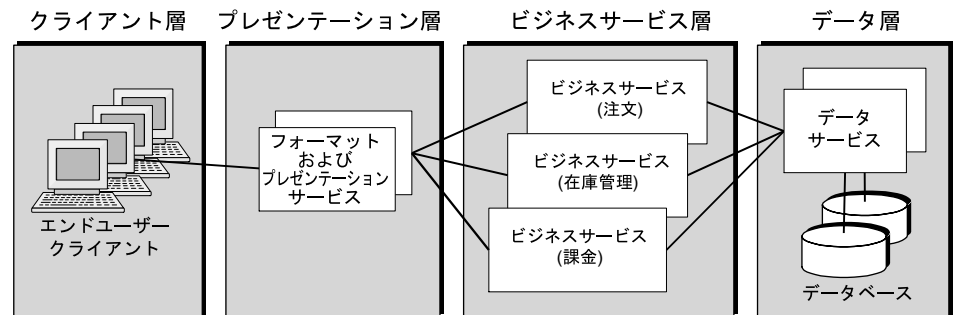
| コンポーネント | 依存するサーバー | サポートするサーバー |
|------------------|----------|---|
| Directory Server | なし | Portal Server Calendar Server Messaging Server Instant Messaging Access Manager |

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。それらの層は、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。

次の図は、ソリューションアーキテクチャの論理層の次元を示しています。

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



大体、論理層アーキテクチャは、19 ページの図 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。35 ページの「インフラストラクチャサービスレベル」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 などの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。通常、プレゼンテーション層のロジックは、HTML 形式または XML 形式での配信のためにデータを準備したり、処理の要求を受信したりする 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 コンポーネントで構成されます。プレゼンテーション層には、ビジネスサービス層の **ビジネスサービス** に対する、個人向けにカスタマイズされ、セキュリティ保護されたアクセスを提供できるポータルサービスが含まれることがあります。
- **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な機能を実行するロジックで構成されます。これらの機能には、データの処理、ビジネスルールの実装、複数のユーザーの調整、データベースや旧バージョンの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、ビジネスサービス層は、Java オブジェクト、EJB コンポーネント、メッセージ駆動型 Beans などのように、J2EE 分散型コンポーネントモデルに準拠している、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およびサービスの構成部品は、SOAP (Simple Object Access Protocol) インタフェース標準に準拠し、サービス指向型アーキテクチャモデル内で緩やかに結合した **Web サービス** としてカプセル化することができます。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スタンドアロン**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

41 ページの図 2-4 に示したアーキテクチャの次元では、4 つの層で表される、コンポーネントの論理的および物理的な独立性が強調されています。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：** アーキテクチャモデルの 4 つの層は論理的な独立性を表しています。つまり、1 つの層 (たとえば、ビジネスサービス層) のアプリケーションロジックを、他の層のロジックとは関係なく変更できます。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。異なる層のロジックは異なるハードウェアプラットフォーム上に (たとえば、プロセッサ構成、チップセット、オペレーティングシステム) 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。

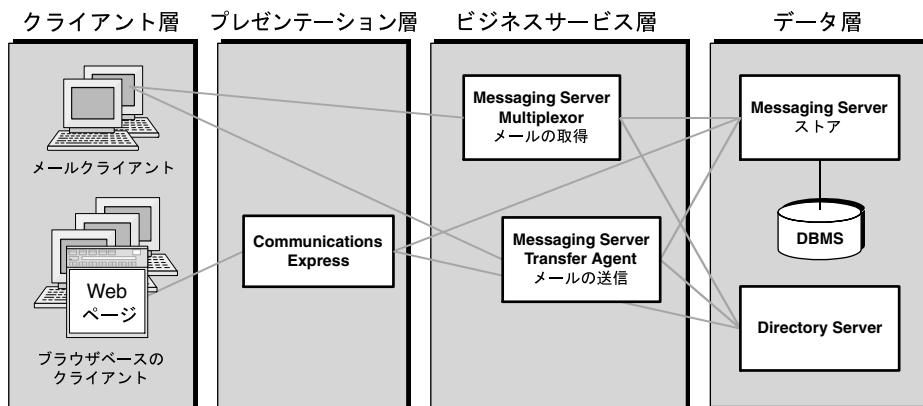
アプリケーションコンポーネントまたはインフラストラクチャコンポーネントをハードウェア環境 (つまり、配備アーキテクチャ) にマッピングする方法は、ソフトウェアソリューションの規模および複雑さに応じて、多数の要素によって決定されます。小規模の配備の場合は、配備アーキテクチャに含まれるのは数台のコンピュータのみである場合があります。大規模の配備の場合は、ハードウェア環境へのコンポーネントのマッピングには、各コンピュータの速度と演算能力、ネットワークリンクの速度と帯域幅、セキュリティおよびファイアウォールの考慮事項、高可用性およびスケラビリティのためのコンポーネントのレプリケーションの方針などの要素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

39 ページの図 2-3 に示すように、Java ES インフラストラクチャ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本となるインフラストラクチャ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 Messaging Server の論理的に区別された設定を使用して実装されます。これらの区別された設定は、それぞれ異なるサービスセットを提供します。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の例



注 図 2-3 は、簡略化のためにいくつかの Java ES コンポーネントが省略され、完全な論理アーキテクチャを表しているわけではありません。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。物理的な分離により、サービス品質の要件 (「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の 2 つの次元 (インフラストラクチャ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せん。 |
| 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を変更する必要はありません。 |
| 潜在的な容量 | 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コンポーネントが物理環境にどのように配備されるかに、大きな影響があります。

配備アーキテクチャに影響を与えるサービス品質は、密接に関連しています。1つのシステムの質に関する要件は、他のサービス品質に関する設計に影響を与えることがあります。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。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。

23 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動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 スケーラビリティ |
| Directory Proxy Server | セキュリティ 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 スケーラビリティ |

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 (続き)

| コンポーネント | 影響を受けるシステム品質 |
|---------------------------------|-----------------|
| Sun Cluster | 可用性 スケーラビリティ |
| Sun Remote Services Net Connect | 保守性 |

Sun Cluster ソフトウェア

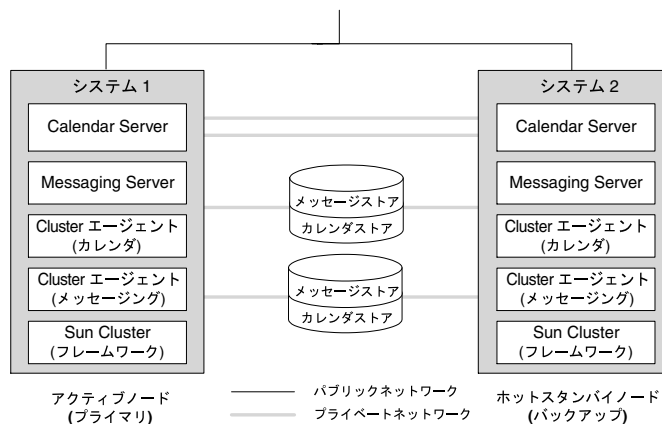
Sun Cluster ソフトウェアは、Java ES コンポーネントおよび Java ES インフラストラクチャによってサポートされるアプリケーションに対する高可用性サービスおよびスケーラビリティサービスを提供します。

クラスタは、緩やかに結合したコンピュータのセットであり、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 つのノードからなるクラスタを次の図に示します。

図 2-6 Sun Cluster ノードを使用した可用性の設計



Sun Cluster データサービスパッケージ (Sun Cluster エージェントとも呼ばれることがあります) 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。

ほかの Java ES コンポーネントから提供されるサービスとは異なり、Sun Cluster 可用性サービスは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。

アーキテクチャの 3 つの次元の統合

図 2-1 に示し、前の節で説明したアーキテクチャの 3 つの次元を統合すると、分散型ソフトウェアソリューションを設計するための 1 つのフレームワークになります。3 つの次元 (インフラストラクチャ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に含まれるコンポーネントは、アーキテクチャ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とソリューションの配備アーキテクチャの設計に影響します。

図 2-1 に示したアーキテクチャのフレームワークに基づいた詳細な設計方法の概要は、このマニュアルでは扱っていません。ただし、3 次元のアーキテクチャフレームワークは、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、Java Enterprise System に含まれるコンポーネントを使用して、開発せず、ほとんど追加設定なしで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で使用します。

ここでは、前の節のアーキテクチャの概念に基づいた、Java Enterprise System がどのように追加設定が不要でそのまま使用できるソリューションをサポートするかを示す例を挙げます。

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てあります。

表 2-4 ビジネス要件の要約: 通信のシナリオ

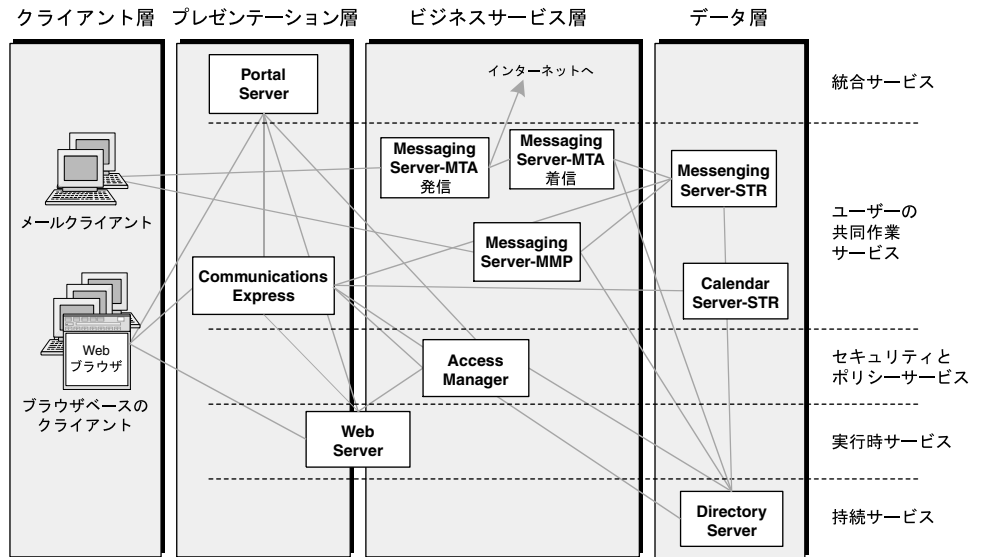
| ビジネス要件 | 説明 | 必要な Java ES サービス |
|------------------|--|------------------|
| シングルサインオン | セキュリティ保護されたエンタープライズリソースおよびサービスへのアクセス (Web アクセスはシングルサインオンの単一の ID に基づく)。 | アイデンティティサービス |
| メッセージング カレンダー | 従業員と外部との電子メールメッセージング。 電子的な従業員用のカレンダー機能や会議の設定。 | 通信サービスと共同作業サービス |
| ポータルアクセス | 電子メール、カレンダー、内部 Web ページな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

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を次の図に示します。このアーキテクチャは、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



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、Portal Server およびプレゼンテーション層のその他の Web ベースのコンポーネントのためのシングルサインオン認証および承認の仲介を行います。Messaging Server コンポーネントには、データ層内のメッセージストア (Messaging Server-STR)、ビジネスサービス層内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、今度は、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、『Java Enterprise System Deployment Planning Guide』を参照してください。

シナリオ例の配備アーキテクチャ

論理アーキテクチャから配備アーキテクチャに移行する際には、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、『Java Enterprise System Deployment Planning Guide』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。

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ているサービス。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。

アーキテクチャ：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) の論理的、物理的な構築ブロック、およびこれらの相互の関係を示す設計。**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と**配備アーキテクチャ**の両方が含まれます。

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**または他のアプリケーションコンポーネントに提供するために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。

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。

サーバー：外部インタフェースを通してサービスにアクセスする**クライアント**に**分散型サービス**または関連する一連のサービスを提供する、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。

配備アーキテクチャ：**論理アーキテクチャ**の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。

ビジネスサービス：複数のクライアントの代わりにビジネスロジックを実行する、つまりマルチスレッド対応プロセスを実行する**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の構成部品。**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の構成部品またはスタンドアロン**サーバー**がビジネスサービスになる場合もあります。

論理アーキテクチャ：分散型アプリケーションの論理的な構築ブロック、およびこれらの構築ブロック間の関係 (またはインタフェース) を示す設計。論理アーキテクチャには、**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**サービス**の両方が含まれます。

この章の重要な用語

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、別のインフラストラクチャ製品を手動で統合する場合と比較した、Java Enterprise System を使用した場合の利点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 の統合インストーラ」
- 58 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」
- 62 ページの「この章の重要な用語」

Java Enterprise System の統合インストーラ

Java ES のすべてのコンポーネントは、1つのインストーラを使用し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 のインストーラは、Java ES ソフトウェアを1つ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 コンポーネントによっては、インストール時の設定もいくつか可能にします。

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。また、英語の他に、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、および日本語の7つの言語をサポートしています。

ここでは、Java ES の統合インストーラに関する次の点について説明します。詳細については、『Java Enterprise System インストールガイド』を参照してください。

- [既存のソフトウェアのチェック](#)
- [依存性の確認](#)
- [初期設定](#)
- [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す。

同様に、インストーラは J2SE や NSS などの Java ES 共有コンポーネント (25 ページの「共有コンポーネント」を参照) がインストールされているかどうかを確認します。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

インストールしたコンポーネントが相互に正常に機能するように、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サービス

Java Enterprise System の重要な機能として、ユーザーアイデンティティの統合管理、統合された認証、および承認フレームワークがあります。

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。

- [単一アイデンティティ](#)
- [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。ユーザーは、この[単一アイデンティティ](#)に基づいて、ポータルなどの各種のリソース、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。

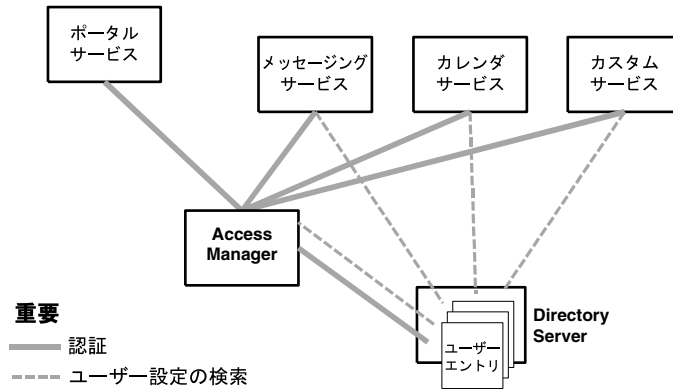
この統合されたアイデンティティとセキュリティ機能は、Directory Server、Access Manager、およびその他の Java ES コンポーネント間の密接な共同作業に基づいています。

Java ES のサービスやリソースにユーザーがアクセスできるのは、ユーザーリポジトリまたは[ディレクトリ](#)のシングルユーザーエントリにユーザー固有の情報が格納されているためです。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 では、ユーザーエントリは Directory Server によって作成されるディレクトリに格納されます。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。Java Enterprise System から提供される、この強力な機能は「[シングルサインオン](#)」と呼ばれます。

認証とシングルサインオン

Java ES の認証サービスおよび承認サービスは、Access Manager から提供されます。Access Manager は、Directory Server の情報を使用して、Java ES の Web サービスまたは企業内におけるその他の Web ベースのサービスとユーザー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 などの一部の Java ES コンポーネントでは、ポリシーエージェントの機能は Access Manager サブコンポーネントによって提供されています ([74 ページの「Sun Java System Access Manager 6 2005Q1」](#)を参照)。

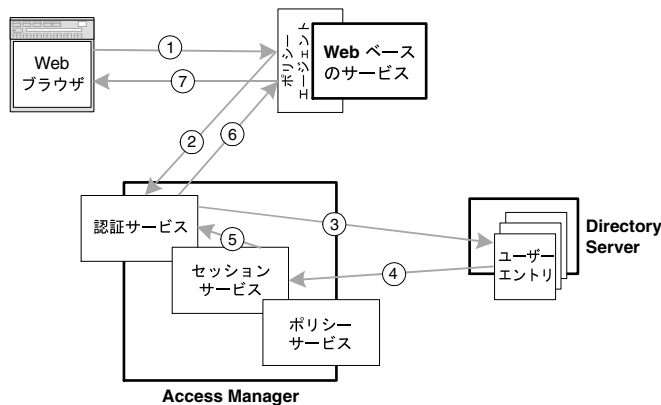
認証

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。

図 3-2 に、認証が行なわれる順序を示します。ポリシーエージェントは、電話帳へのログオン要求を仲介し (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します (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)。

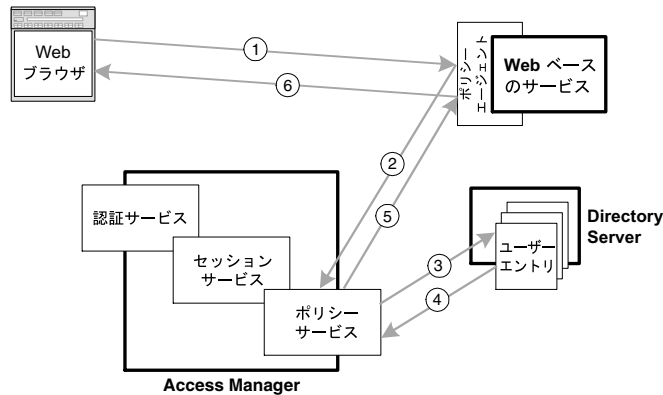
認証されたユーザーがセキュリティ保護された別のサービスにアクセスしようとする時、ブラウザはセッショントークンを対応するポリシーエージェントに渡します。ポリシーエージェントはセッションサービスを使用して、そのユーザーの以前の認証が現在も有効であることを確認します。これにより、ユーザー ID およびパスワードの再入力を求めずに、そのユーザーに対して別のサービスへのアクセスが許可されます。

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が記述されたルールです。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。

ポリシーは、Access Manager のポリシーエージェントによって適用されます。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。

シングルサインオン：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、同じシステム内のほかのサービスに自動的に適用できるようにする機能。

単一アイデンティティ：ユーザーが **Java Enterprise System** ディレクトリに 1 つのユーザーエントリを持つためのアイデンティティ。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。

ディレクトリ：データの書き込みよりもデータの読み込み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が記述されたルール。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。

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

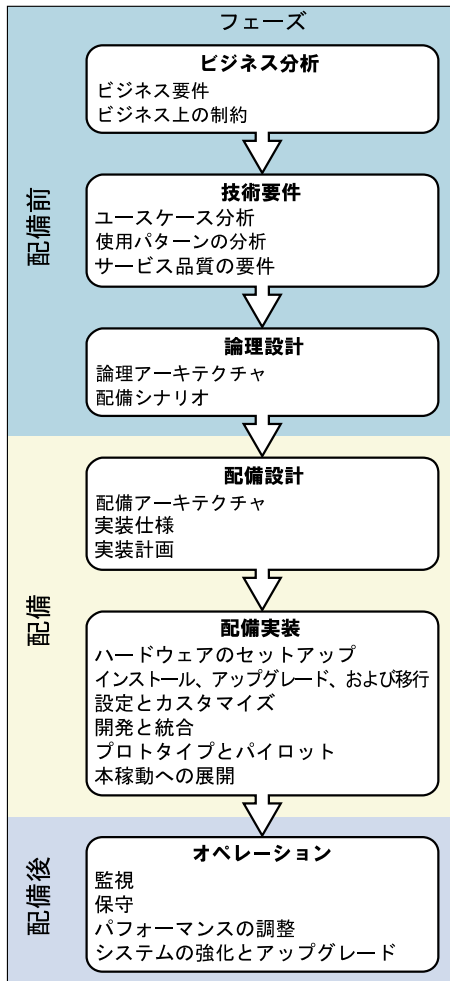
Java ES ソリューションのライフサイクルについては、Java ES ソフトウェアを使用するビジネスソリューションを実装するための標準的な方法として、[第 1 章「Java Enterprise System の概要」](#)で説明しました。この章では、ライフサイクルの各フェーズで行う作業について説明します。簡単に参照できるようにライフサイクルの図を [64 ページの図 4-1](#) に再度示してあります。

この章では、各フェーズに関連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」](#)
- [72 ページの「この章の重要な用語」](#)

図 4-1 ソリューションのライフサイクルの作業



配備前

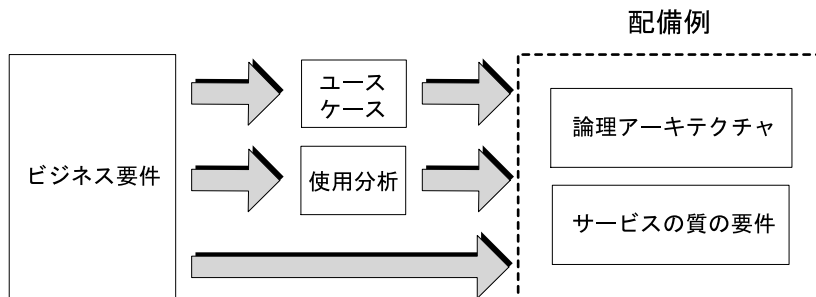
ライフサイクルの**配備前**のフェーズでは、ビジネスのニーズに関する分析に基づいて**配備シナリオ**を作成します。配備シナリオは、配備を設計するための仕様としての役割を果たします。

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を使用して、作成予定のソフトウェアシステムとユーザーの対話をモデル化する**ユースケース**を作成します。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を使用して、提案する配備が満たす必要のあるサービス品質の要件 (**45 ページの表 2-2** を参照) を明確にします。
- 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「Java Enterprise System ソリューションアーキテクチャ」** で説明した概念に基づいて、論理アーキテクチャを設計します。論理アーキテクチャは、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に関する要件と組み合わせて、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フェーズの詳細については、『**Java Enterprise System Deployment Planning Guide**』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**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、これを本稼動環境に実装、テスト、および展開します。

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、ソリューションをサポートするために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 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。

- 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。
- 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、および本稼動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

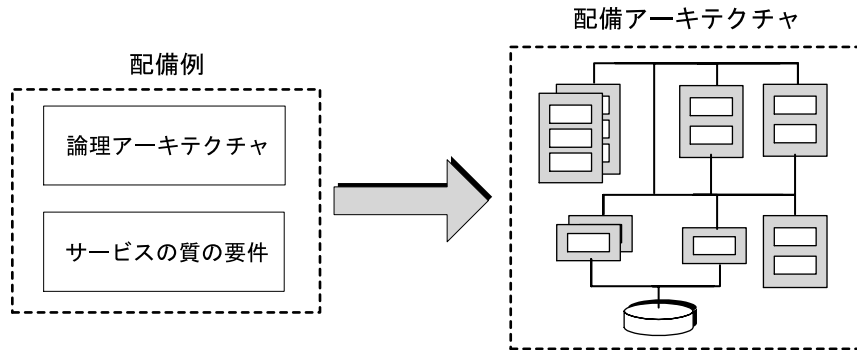
配備設計フェーズでは、上位レベルの配備アーキテクチャを作成し、その後に下位レベルの実装仕様を作成します。

配備アーキテクチャ

配備アーキテクチャは、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、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、配備設計の開始時に、Java Enterprise Systemは一連の[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または複数の参照アーキテクチャ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を調整したり、参照アーキテクチャ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、前の節および図 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、『Java Enterprise System 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、次のような物理環境のすべての詳細情報が含まれています。たとえば、コンピュータやネットワーク設計のほかに、ケーブル、スイッチ、ルータ、ロードバランサなどのネットワークハードウェアや記憶装置などを指定します。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、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用して、配備アーキテクチャ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できます (55 ページの「Java Enterprise System の統合インストーラ」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。ユーザーが各種の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 (58 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、要件どおり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、通常ポータルチャンネルを提供するには Portal Server を、承認タスクを実行するには Access Manager を、ウイルスチェックおよびスパム防止フィルタ処理を使用するには Messaging Server をカスタマイズする必要があります。

開発と統合

通常、配備シナリオに指定された論理アーキテクチャによって、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定されます。

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、データやアプリケーションコードを移行する必要がある場合があります。

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。

カスタム開発のサービスが比較的少ない（つまりほとんど追加設定の必要がない配備の場合）は、ソリューションに必要なのは 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だけであることがあります。

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の欠点が判明した場合は、アーキテクチャ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、本稼動環境で配備し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**: 新しい機能の追加や Java ES 以外のコンポーネントの交換のためにシステムに新しい Java ES コンポーネントを追加します。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。

開発 : Java Enterprise System ソリューション開発プロセスの作業の 1 つ。この作業により **配備アーキテクチャ** のカスタムコンポーネントが作成され、テストされます。

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 1 つの段階。配備段階では、配備シナリオに基づいて配備設計を作成し、本稼動環境に実装、プロトタイプ化、および展開します。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。

配備後 : Java Enterprise System ソリューションのライフサイクルプロセスの 1 つの段階。この段階では、分散型アプリケーションの起動、監視、パフォーマンス最適化のための調整、および動的なアップグレードによる新しい機能の組み入れが行われます。

配備シナリオ : ビジネスのニーズを満たすためにソリューションに要求される、Java Enterprise System ソリューションの **論理アーキテクチャ** および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、スケーラビリティ、保守性、および潜在的な容量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。

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 1 つの段階。この段階では、ビジネスのニーズに基づいて **配備シナリオ**、つまり **論理アーキテクチャ** およびソリューションが満たす必要があるサービス品質に関する一連の要件を作成します。

ユースケース : エンドユーザーが **分散型のエンタープライズアプリケーション** で実行する特定のタスク、またはタスクのセット。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本情報として使用されます。

リファレンス配備アーキテクチャ : パフォーマンス用に設計、実装、およびテストされた **配備アーキテクチャ**。リファレンス配備アーキテクチャは、カスタムソリューションの配備アーキテクチャを設計するときの開始点として使用されます。

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。

- **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サービスを提供します。システムサービスコンポーネントには、18 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、可用性サービスが含まれます。
- **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。
- **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。

コンポーネントのマニュアルを探すには、『Java Enterprise System ドキュメントロードマップ』(<http://docs.sun.com/doc/819-1912?l=ja>) 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サービスを提供します。次の節で 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。

- [Sun Java System Access Manager 6 2005Q1](#)
- [Sun Java System Application Server Enterprise Edition 8 2005Q1](#)
- [Sun Java System Calendar Server 6 2005Q1](#)
- [Sun Java System Directory Server 5 2005Q1](#)
- [Sun Java System Instant Messaging 7 2005Q1](#)
- [Sun Java System Message Queue 3 2005Q1](#)
- [Sun Java System Messaging Server 6 2005Q1](#)
- [Sun Java System Portal Server 6 2005Q1](#)
- [Sun Java System Web Server 6.1 2005Q1](#)

Sun Java System Access Manager 6 2005Q1

Sun Java System Access Manager (Access Manager) は、企業が 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 の保守プロセスを管理するためのインフラストラクチャを提供します。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 (83 ページの「[Sun Java System Delegated Administrator 6 2005Q1](#)」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 2005Q1

Sun Java System Application Server (Application Server) は、アプリケーションサービスと Web サービスの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。

- **ドメイン管理サーバー**： 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 のインストールとホストするアプリケーションの管理および設定に使用する、グラフィカル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 のインストールとホストするアプリケーションの管理および設定に使用する、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 2005Q1

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 2005Q1

Sun Java System Directory Server (Directory Server) が提供するディレクトリサービスを利用すると、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を一元管理できます。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の統合化に対応した集中的のリポジトリとして動作します。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 2005Q1

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することが可能になります。Instant Messaging は、イントラネットとインターネットの両方で活用でき、他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 2005Q1

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に対応した標準ベースのソリューションです。Message Queue は、オープン標準規格である JMS (Java Message Service) を実装する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 メッセージ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):** マルチブローカ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 2005Q1

Sun Java System Messaging Server (Messaging Server) は、企業および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:** メッセージの格納と取得を可能にします。

- **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 2005Q1

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 は個人用のカスタマイズ、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 をインストール可能な単一のコンポーネントとして提供します。

Sun Java System Web Server 6.1 2005Q1

Sun Java System Web Server (Web Server) は、オープン標準規格に基づいて構築された、マルチプロセスかつマルチスレッド対応のセキュリティ保護された Web サーバーです。また、Web Server は、あらゆる規模の企業に適用可能な高いパフォーマンス、信頼性、スケーラビリティ、および管理能力を提供します。Web Server では、JDK 1.4.1、Java サブレット 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されます。

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。

- [Sun Cluster 3.1 9/04 と Sun Cluster エージェント](#)
- [High Availability Session Store 2005Q1](#)

Sun Cluster 3.1 9/04 と Sun Cluster エージェント

Sun Cluster ソフトウェアは、Java Enterprise System および Java ES インフラストラクチャに基づいたアプリケーションに対する高可用性サービスおよびスケラビリティサービスを提供します。

クラスタは、緩やかに結合したコンピュータ (クラスタノード) のセットであり、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、内部の冗長性を利用して、障害の発生時にもクラスタリソースへのほぼ連続する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、個別の CD に収録されています。

High Availability Session Store 2005Q1

Sun Java System High Availability Session Store (HADB) は、障害の発生時にもアプリケーションのデータが利用可能になるようにデータストアを提供します。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。

Application Server、Access Manager、および Message Queue 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。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。

- [Sun Java System Communications Express 2005Q1](#)
- [Sun Java System Connector for Microsoft Outlook 6 2005Q1](#)
- [Sun Java System Directory Proxy Server 5 2005Q1](#)
- [Sun Java System Portal Server, Secure Remote Access 6 2005Q1](#)

Sun Java System Communications Express 2005Q1

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、カレンダー、アドレス帳、およびメールの 3 つのクライアントモジュールで構成されます。Communications Express は、Messaging Server または Calendar Server、あるいはこ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 v.1 (Schema 1) または 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 Connector for Microsoft Outlook 6 2005Q1

Sun Java System Connector for Microsoft Outlook を使用すると、Sun Java Enterprise System とともに Outlook を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 では、このモデルを使用して、Messaging Server のメールおよび Calendar Server のカレンダー情報という 2 つの別個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。

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 2005Q1

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ミッションクリティカルなディレクトリサービスの中心的なコンポーネントです。Directory Proxy Server は、アプリケーション層のロードバランサーおよびフェイ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する、LDAP アプリケーション層のプロトコルゲートウェイです。

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 2005Q1

Sun Java System Portal Server, Secure Remote Access (Portal Server, Secure Remote Access) は、Portal Server の機能を拡張し、リモートのブラウザから Portal Server コンテンツおよびサービスにセキュリティ保護されたリモートアクセスができるようにします。ユーザーは、クライアントソフトウェアを用意する必要はありません。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。

- [Sun Java System Administration Server \(およびコンソール \) 5 2005Q1](#)
- [Sun Java System Directory Preparation Tool 2005Q1](#)
- [Sun Java System Delegated Administrator 6 2005Q1](#)
- [Sun Remote Services Net Connect 2005Q1](#)

Sun Java System Administration Server (およびコンソール) 5 2005Q1

Sun Java System Administration Server とサーバーコンソールは、Directory Server、Messaging Server、および Directory Proxy Server を管理するためのグラフィカルツールを提供します。管理サーバーは、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管理サーバー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管理サーバーは、まとめてインストール可能な 1 つのコンポーネントとして提供されます。

Sun Java System Directory Preparation Tool 2005Q1

Sun Java System Directory Preparation Script を使用して、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で Directory Server を設定します。

Sun Java System Delegated Administrator 6 2005Q1

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 (comadmin) 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。

Sun Remote Services Net Connect 2005Q1

Sun Remote Services Net Connect は、ユーザーの IT 環境の制御を支援するために設計されたシステム管理サービスのコレクションです。Web で配信されるこれらのサービスを使用すると、システムを自己監視したり、パフォーマンスレポートやトレンドレポートを作成したり、システムイベントの通知を自動的に受信することによって、実際の問題が発生する前に潜在的な問題の管理にすばやく対応できるようになります。

Java ES インストーラでは、Sun Remote Services Net Connect 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。

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 サーバー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 ベースの構築ツール)
- Apache SOAP (Simple Object Access Protocol) Runtime
- 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) プラットフォーム 5.0
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Enterprise Web Application Framework)
- JavaHelp™ Runtime
- JavaMail™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXM (Java API for XML Messaging) Client 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
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- NSPERL を含む Perl LDAP
- SAAJ (SOAP with Attachments API for Java)
- SAML (Security Assertions Markup Language)
- SASL (Simple Authentication and Security Layer)
- SNMP (Simple Network Management Protocol) Peer
- Sun Explorer Data Collector (Solaris のみ)
- Sun Java Monitoring Framework
- Sun Java Web Console
- Tomcat Servlet JSP Container
- XML C Library (libxml)
- WSCL (Web services Common Library)

索引

A

Access Manager

インフラストラクチャサービスとして, 39
システムサービスコンポーネント, 21
説明, 74

Administration Server

サービス品質コンポーネント, 24
説明, 82

Ant (Jakarta ANT Java/XML ベースの構築ツール),
83

Apache SOAP Runtime, 83

Application Server

インフラストラクチャサービスとして, 39
システムサービスコンポーネント, 21
説明, 75

B

Berkeley DB, 83

C

Calendar Server

インフラストラクチャサービスとして, 39
システムサービスコンポーネント, 22
説明, 76

CAPI (Java Calendar API), 84

Communications Express

サービス品質コンポーネント, 24
説明, 80

Connector for Microsoft Outlook

サービス品質コンポーネント, 24
説明, 81

D

Delegated Administrator

サービス品質コンポーネント, 24
説明, 83

Directory Preparation Tool

サービス品質コンポーネント, 24
説明, 83

Directory Proxy Server

サービス品質コンポーネント, 24
システムコンポーネントとして, 46
説明, 81

Directory Server

インフラストラクチャサービスとして, 39
システムサービスコンポーネント, 22
説明, 76

E

EJB コンポーネント, 42

H

High Availability Session Store

サービス品質コンポーネント, 23
説明, 80

I

ICU (International Components for Unicode), 83

Instant Messaging

インフラストラクチャサービスとして, 39
システムサービスコンポーネント, 22
説明, 76

J

J2EE

コンポーネント, 42
プラットフォーム, 22
分散型コンポーネントモデル, 42

J2ME プラットフォーム, 42

J2SE (Java 2 Platform, Standard Edition), 25, 84

JAF (JavaBeans™ Activation Framework), 84

JATO (Java Studio Enterprise Web Application Framework), 84

JavaHelp, 84

JavaMail, 84

Java サブレットコンポーネント, 42

JAXB (Java Architecture for XML Binding), 84

JAXM (Java API for XML Messaging), 84

JAXP (Java API for XML Processing), 84

JAXR (Java API for XML Registries), 84

JAX-RPC, 84

JDMK (Java Dynamic Management™ Kit), 84

JMS (Java Message Service), 22

JSP コンポーネント, 42

JSS (Java Security Services), 25, 84

K

KT Search Engine (KTSE), 84

L

LDAP, 42, 62

LDAP C SDK, 84

LDAP Java SDK, 84

Linux, 39

M

Message Queue

インフラストラクチャサービスとして, 39
システムサービスコンポーネント, 22
説明, 77

Messaging Server

インフラストラクチャサービスとして, 39
システムサービスコンポーネント, 22
説明, 77

N

NSPR (Netscape Portable Runtime), 25, 84

NSS (Network Security Services), 25, 84

P

Perl LDAP, 84

Portal Server

インフラストラクチャサービスとして, 39
システムサービスコンポーネント, 22
説明, 78

Portal Server, Secure Remote Access

サービス品質コンポーネント, 24
システムコンポーネントとして, 46

説明, 81

R

Remote Services Net Connect

サービス品質コンポーネント, 25

説明, 83

S

SAAJ (SOAP with Attachments API for Java), 84

SAML (Security Assertions Markup Language), 84

SASL (Simple Authentication and Security Layer),
84

SNMP (Simple Network Management Protocol), 84

SOAP (Apache), 83

Solaris, 39

Sun Cluster

エージェント, 48

可用性サービスとして, 47

サービス品質コンポーネント, 23

説明, 79

Sun Explorer Data Collector, 84

Sun Java Monitoring Framework, 84

Sun Java System 製品

Access Manager、「Access Manager」を参照

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

Sun Java Web Console, 84

T

Tomcat Servlet JSP Container, 84

W

Web Server

インフラストラクチャサービスとして, 39

システムサービスコンポーネント, 22

説明, 78

Web サービス, 19

J2EE コンポーネント, 42

定義, 53

WSCL (Web services Common Library), 84

X

XML C Library, 84

あ

アーキテクチャ
参照, 67

次元、「アーキテクチャの次元」を参照

紹介, 33

ソリューション, 34

定義, 53

配備, 66

論理, 65

アーキテクチャの次元

インフラストラクチャサービスの依存関係, 35

サービスの品質, 44

統合, 48

論理層, 41

アイデンティティ

管理, 58

サービス, 19, 58

シングルユーザー, 58

アクセスコンポーネント

概要, 23

説明, 80

アクセスサービス, 20

アップグレード、「導入シナリオ」を参照

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, 53

論理層アーキテクチャ, 41

アプリケーションサービス, 19, 36

アンインストーラ, 57

い

移行、Java ES 導入シナリオ, 29

依存, 39, 57

依存性の確認、インストーラ, 56

インストールされているソフトウェアの検出, 56

インフラストラクチャ

サービスの依存関係、「分散型のサービス」を参
照

分散型のエンタープライズアプリケーションの,

18

え

エンドユーザー

定義, 31

分散型アプリケーション, 18

お

オペレーティングシステムサービス, 36

か

開発

定義, 72

配備作業, 70

拡張、「導入シナリオ」を参照

可用性

サービス, 47, 79

要件, 45, 46, 47

可用性コンポーネント

概要, 23

説明, 79

可用性サービス, 19

管理コンポーネント

概要, 24

説明, 82

管理サービス, 20

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, 19

共有コンポーネント, 83

概要, 25

定義, 31

く

クライアント

システムサービスコンポーネント, 21

定義, 53

クライアント論理層, 42

クラスタ、「Sun Cluster」を参照

け

言語サポート, 56

こ

コンポーネント

EJB, 42

J2EE, 42

JSP, 42

アクセス, 23

依存, 39

インストールされているバージョンの検出, 56

インフラストラクチャサービス, 38

可用性, 23

管理, 24

共有, 25, 83

サービス品質, 23, 79

サーブレット, 42

システムサービス, 21, 74

システム、「システムコンポーネント」を参照

定義, 31

分散型, 18

さ

サーバー

スタンドアロン, 42

定義, 53

サービス

Web, 42

インフラストラクチャ, 18

インフラストラクチャ、「分散型のインフラストラクチャサービス」を参照

高可用性, 47, 79

スケーラビリティ, 47, 79

定義, 31

ビジネス, 42

サービス品質コンポーネント

概要, 23

説明, 79

定義, 31

サービス品質の要件

可用性, 45, 46, 47

スケーラビリティ, 45, 46, 47

セキュリティ, 45, 46

潜在的な容量, 45

パフォーマンス, 45

保守性, 45, 47

作業、Java Enterprise System, 25, 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, 18

設定, 57

システムコンポーネント

基本情報, 20

共有コンポーネント, 25, 83

サービス品質コンポーネント, 23

システムサービス, 74

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, 31

- システムサービス
 - 基本情報, 21
 - 定義, 31
- システムサービスコンポーネント
 - 依存, 39
 - 概要, 21
 - 定義, 32
- 持続サービス, 37
- 実行時サービス, 37
- 実装仕様, 68
- シナリオ、「配備シナリオ」を参照
- 承認, 61
- シングルサインオン
 - Java ES の機能, 21, 59
 - インフラストラクチャサービスレベル, 37
 - 実装, 60
 - 定義, 62

す

- スケーラビリティ
 - サービス, 47, 79
 - 要件, 45, 46, 47

せ

- セキュリティ
 - サービス, 19
 - ポリシーサービス, 37
 - 要件, 45, 46
- 潜在的な容量の要件, 45

そ

- 層、論理
 - アプリケーションのアーキテクチャ, 41
 - クライアント, 42
 - データ, 42

- ビジネスサービス, 42
- プレゼンテーション, 42
- ソリューション、Java ES
 - アーキテクチャ, 33
 - カスタムおよびアウトオブボックス, 49
 - ライフサイクル, 26
 - 例, 49

た

- 単一アイデンティティ
 - 概要, 58
 - 定義, 62

つ

- 通信サービス, 19

て

- ディレクトリ
 - 定義, 62
 - ユーザーデータストア, 58

と

- 統合
 - Java ES 導入シナリオ, 30
 - 機能、「統合機能」を参照
 - サービス, 37
- 統合機能
 - アイデンティティとセキュリティ, 20, 58
 - 概要, 20
 - 共有コンポーネント, 20
 - 統合インストーラ, 20, 55
- 導入シナリオ、Java ES
 - 新しいシステム, 29

- アップグレード, 29
- 拡張, 29
- 基本情報, 29
- 強化, 29
- 定義, 32

トレーニング、Java ES 導入シナリオ, 30

に

認証, 59

ね

ネットワークトランスポートサービス, 37

は

ハードウェア、Java ES 導入シナリオ, 30

配備

- アーキテクチャ, 66
- 開発とカスタマイズ, 70
- 参照アーキテクチャ, 67
- 実装, 69
- 設計, 66
- 定義, 72
- プロトタイプテスト, 70
- 本稼動への展開, 71
- ライフサイクルの段階, 66
- ライフサイクルのフェーズ, 66

配備アーキテクチャ

- 紹介, 33
- 設計, 66
- 層によるアーキテクチャに対する関係, 43
- 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, 45

ひ

ビジネスサービス

- 定義, 53
- プレゼンテーション層, 42

ふ

プラットフォームサービス, 36

プロトタイプ化, 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、「分散型のサービス」を参照

分散型のエンタープライズアプリケーション
インフラストラクチャ, 18

- 基本情報, 18
- 定義, 32

分散型のサービス

- Web, 19
- アイデンティティ, 19
- アクセス, 20
- アプリケーションレベル, 36
- インフラストラクチャ, 18

概要, 18

可用性, 19

管理, 20

持続, 37

実行時, 19, 37

セキュリティ, 19, 37

通信と共同作業, 19

統合, 37

ネットワークトランスポート, 37

- プラットフォーム, 36
- ポータル, 19
- ミドルウェア, 36
- メッセージング, 37
- ユーザーの共同作業, 37

ほ

- ポータルサービス, 19
- 保守性の要件, 45, 47
- ポリシー
 - 承認, 61
 - 定義, 62
- 本稼動への展開, 71

ま

- マニュアルセット、Java ES, 12

み

- ミドルウェアサービス, 36

め

- メッセージングサービス, 37

ゆ

- ユーザー、「エンドユーザー」を参照
- ユーザーエントリ, 28
- ユーザーカテゴリ
 - IT マネージャ, 28
 - システムアナリスト, 28
 - システムインテグレータ, 28

- システム管理者, 28
- 設計者, 28
- 代行管理者, 28
- 特化したシステム管理者, 28
- ビジネスプランナー, 28
- フィールドエンジニア, 28
- ユーザーの共同作業サービス, 37
- ユーザーのプロビジョニング, 68
- ユーザープロファイル, 28
- ユースケース
 - 概要, 65
 - 定義, 72

ら

- ライフサイクルのフェーズ
 - 配備, 26, 66
 - 配備後, 26, 71
 - 配備前, 26, 65

り

- リファレンス配備アーキテクチャ
 - 概要, 67
 - 定義, 72

ろ

- 論理アーキテクチャ
 - インフラストラクチャサービスレベル, 36
 - 基本情報, 65
 - 紹介, 33
 - 定義, 53
 - 例, 51