

Cosminexus 概説

概説書

3020-3-M01-60

マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，
巻末の「ソフトウェアマニュアルのサービス ご案内」をご参
照ください。

対象製品

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 (x64) , Windows Server 2003 R2 (x64)

P-2443-7D74 uCosminexus Application Server Standard 07-60

P-2443-7K74 uCosminexus Application Server Enterprise 07-60

P-2443-7M74 uCosminexus Web Redirector 07-60

P-2443-7S74 uCosminexus Service Platform 07-60

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Vista , Windows XP

P-2443-7E74 uCosminexus Developer Standard 07-60

P-2443-7F74 uCosminexus Developer Professional 07-60

P-2443-7T74 uCosminexus Service Architect 07-60

P-2443-7U74 uCosminexus Operator 07-60

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 (x64) , Windows Server 2003 R2 (x64) , Windows Vista , Windows XP

P-2443-7H74 uCosminexus Client 07-60

適用 OS : AIX 5L V5.2 , AIX 5L V5.3

P-1M43-7D71 uCosminexus Application Server Standard 07-60

P-1M43-7K71 uCosminexus Application Server Enterprise 07-60

P-1M43-7M71 uCosminexus Web Redirector 07-60

P-1M43-7S71 uCosminexus Service Platform 07-60

適用 OS : HP-UX 11i V2 (IPF) , HP-UX 11i V3 (IPF)

P-1J43-7D71 uCosminexus Application Server Standard 07-60

P-1J43-7K71 uCosminexus Application Server Enterprise 07-60

P-1J43-7M71 uCosminexus Web Redirector 07-60

適用 OS : Red Hat Enterprise Linux AS 3 (x86) , Red Hat Enterprise Linux AS 4 (x86) , Red Hat Enterprise Linux ES 3 (x86) , Red Hat Enterprise Linux ES 4 (x86) , Red Hat Enterprise Linux AS 3 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux ES 3 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T)

P-9S43-7D71 uCosminexus Application Server Standard 07-60

P-9S43-7K71 uCosminexus Application Server Enterprise 07-60

P-9S43-7M71 uCosminexus Web Redirector 07-60

適用 OS : Red Hat Enterprise Linux AS 3 (x86) , Red Hat Enterprise Linux AS 4 (x86) , Red Hat Enterprise Linux 5 Advanced Platform (x86) , Red Hat Enterprise Linux ES 3 (x86) , Red Hat Enterprise Linux ES 4 (x86) , Red Hat Enterprise Linux 5 (x86) , Red Hat Enterprise Linux AS 3 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64) , Red Hat Enterprise Linux ES 3 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux 5 (AMD/Intel 64)

P-9S43-7S71 uCosminexus Service Platform 07-60

適用 OS : Red Hat Enterprise Linux AS 3 (IPF) , Red Hat Enterprise Linux AS 4 (IPF) , Red Hat Enterprise Linux 5 Advanced Platform (Intel Itanium)

P-9V43-7D71 uCosminexus Application Server Standard 07-60

P-9V43-7K71 uCosminexus Application Server Enterprise 07-60

P-9V43-7M71 uCosminexus Web Redirector 07-60

適用 OS : Solaris 9 , Solaris 10

P-9D43-7D71 uCosminexus Application Server Standard 07-60

P-9D43-7K71 uCosminexus Application Server Enterprise 07-60

P-9D43-7M71 uCosminexus Web Redirector 07-60

P-9D43-7S71 uCosminexus Service Platform 07-60

印の製品については、サポート時期をご確認ください。

これらのプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」をご確認ください。

本製品では日立トレース共通ライブラリをインストールします。

輸出時の注意

本製品を輸出される場合には、外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上、必要な手続きをお取りください。

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

商標類

AIX は、米国における米国 International Business Machines Corp. の登録商標です。

AMD は、Advanced Micro Devices, Inc. の商標です。

Borland のブランド名および製品名はすべて、米国 Borland Software Corporation 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。

cFramework は、株式会社イーシー・ワンの日本における商標です。

CORBA は、Object Management Group が提唱する分散処理環境アーキテクチャの名称です。

HP-UX は、米国 Hewlett-Packard Company のオペレーティングシステムの名称です。

IIOP は、OMG 仕様による ORB(Object Request Broker) 間通信のネットワークプロトコルの名称です。

Intel は、Intel Corporation の会社名です。

Internet Explorer は、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。

Itanium は、アメリカ合衆国および他の国におけるインテル コーポレーションまたはその子会社の登録商標です。

iWay および iWay Software は、Information Builders, Inc. の米国およびその他の国における登録商標です。

Java 及びすべての Java 関連の商標及びロゴは、米国及びその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。

JDK は、米国 Sun Microsystems, Inc. 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。

Linux は、Linus Torvalds の米国およびその他の国における登録商標あるいは商標です。

Microsoft は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Microsoft Internet Information Services は、米国 Microsoft Corp. の商品名称です。

Microsoft SQL Server は、米国 Microsoft Corp. の商品名称です。

Mozilla は、Mozilla Foundation の、米国およびその他の国における商標です。

MyEclipse は、米国 Genuitec 社の商品名称です。

OMG, CORBA, IIOP, UML, Unified Modeling Language, MDA, Model Driven Architecture は、Object Management Group, Inc. の米国及びその他の国における登録商標または商標です。

ORACLE は、米国 Oracle Corporation の登録商標です。
Oracle は、米国 Oracle Corporation 及びその子会社、関連会社の登録商標です。
Oracle9i は、米国 Oracle Corporation の商標です。
Oracle 10g は、米国 Oracle Corporation の商標です。
PA-RISC は、米国 Hewlett-Packard Company の商標です。
R/3 は、SAP AG のドイツ及びその他の国における登録商標または商標です。
Red Hat は、米国およびその他の国で Red Hat, Inc. の登録商標若しくは商標です。
SAP は、SAP AG のドイツ及びその他の国における登録商標または商標です。
SOAP (Simple Object Access Protocol) は、分散ネットワーク環境において XML ベースの情報を交換するための通信プロトコルの名称です。
Solaris は、米国 Sun Microsystems, Inc. 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。
SQL Server は、米国法人 Sybase, Inc. の商標です。
Sun, Sun Microsystems, Java は、米国 Sun Microsystems, Inc. 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。
UNIX は、X/Open Company Limited 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。
W3C は、World Wide Web Consortium の商標 (多数の国において登録された) です。
Windows は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。
Windows Server は、米国 Microsoft Corporation の米国及びその他の国における登録商標です。
Windows Vista は、米国 Microsoft Corporation の米国及びその他の国における登録商標です。
X/Open は、X/Open Company Limited の英国ならびに他の国における登録商標です。
オブジェクトワークスは、株式会社 野村総合研究所の登録商標です。
Eclipse は、開発ツールプロバイダのオープンコミュニティである Eclipse Foundation, Inc. により構築された開発ツール統合のためのオープンプラットフォームです。
Struts は Apache Software Foundation が運営する Apache Struts Project が公開するサーブレット・JSP の Web アプリケーション構築用のフレームワークです。
This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).
プログラムプロダクト「P-9D43-7D71, P-9D43-7K71, P-9D43-7M71, P-9D43-7S71」には、米国 Sun Microsystems, Inc. が著作権を有している部分が含まれています。
プログラムプロダクト「P-9D43-7D71, P-9D43-7K71, P-9D43-7M71, P-9D43-7S71」には、UNIX System Laboratories, Inc. が著作権を有している部分が含まれています。

HITACHI
Inspire the Next

 株式会社 日立製作所



発行

2006年4月(第1版)3020-3-M01

2007年12月（第4版）3020-3-M01-60

著作権

All Rights Reserved. Copyright (C) 2006, 2007, Hitachi, Ltd.

変更内容

変更内容 (3020-3-M01-60) uCosminexus Application Server Standard 07-60 , uCosminexus Developer Standard 07-60 , uCosminexus Developer Professional 07-60 , uCosminexus Client 07-60 , uCosminexus Application Server Enterprise 07-60 , uCosminexus Web Redirector 07-60 , uCosminexus Service Platform 07-60 , uCosminexus Service Architect 07-60 , uCosminexus Operator 07-60

追加・変更内容	変更箇所
パッチアプリケーション実行環境としての機能を追加した。	1.1 , 1.2.1 , 1.2.3 , 1.3.1 , 2.1.1 , 2.2.2 , 2.3.2 , 2.3.3 , 3.1.1 , 3.1.2 , 3.1.3 , 3.1.4 , 5 章 , 付録 B
Connector 1.5 仕様に準拠したリソースアダプタを使用して非同期通信ができるようにした。	2.3.3 , 4.1.8 , 4.5 , 6.5
Red Hat Enterprise Linux 5 Advanced Platform , および Red Hat Enterprise Linux 5 の説明を追加した。	2.4
Linux (IPF) についての説明を追加した。	2.4
Solaris についての説明を追加した。	2.4 , 8.3
実行環境をセットアップするための機能として , セットアップウィザード機能を追加した。	3.1.1 , 3.1.4 , 4.1.4 , 4.3.2 , 4.3.3 , 4.6 , 9.1 , 9.2.1
開発環境をセットアップするための機能として , 開発環境インスタントセットアップ機能を使用できるようにした。	6.3.1 , 6.6 , 9.1
次の製品の適用 OS に , Red Hat Enterprise Linux 5 Advanced Platform (x86) , Red Hat Enterprise Linux 5 (x86) , Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64) , および Red Hat Enterprise Linux 5 (AMD/Intel 64) を追加した。 • uCosminexus Service Platform	-
次の製品の適用 OS に Linux (IPF) を追加した。 • uCosminexus Application Server Standard • uCosminexus Application Server Enterprise • uCosminexus Web Redirector	-
次の製品の適用 OS に Solaris を追加した。 • uCosminexus Application Server Standard • uCosminexus Application Server Enterprise • uCosminexus Web Redirector • uCosminexus Service Platform	-
HP-UX (PA-RISC) のサポート中止に伴い , この OS に適応していた次の製品の記述を削除した。 • uCosminexus Application Server Standard • uCosminexus Application Server Enterprise • uCosminexus Web Redirector	-

追加・変更内容	変更箇所
次の製品の適用 OS から HP-UX (IPF) を削除した。 ・ uCosminexus Service Platform	-

単なる誤字・脱字などはお断りなく訂正しました。

変更内容 (3020-3-M01-40) uCosminexus Application Server Standard 07-50, uCosminexus Developer Standard 07-50, uCosminexus Developer Professional 07-50, uCosminexus Client 07-50, uCosminexus Application Server Enterprise 07-50, uCosminexus Web Redirector 07-50, uCosminexus Service Platform 07-50, uCosminexus Service Architect 07-50, uCosminexus Operator 07-50

追加・変更内容
Cosminexus で実現できることについての説明を追加した。
複数の実行環境を切り替える機能を追加した。
定義ファイルを出力することによって、構築済みのシステムの移行性を向上させる機能を追加した。
監査ログを出力できるようにした。
データベースの監査証跡機能と連携できるようにした。
構成ソフトウェアとして MyEclipse for Cosminexus を追加した。
対話ワークフローを実現する機能を追加した。また、構成ソフトウェアとして、Cosminexus Service Coordinator Interactive Workflow, WorkCoordinator Definer Version 6 を追加した。
Cosminexus サービスプラットフォームで、ビジネスモデリングツールで出力した BPEL 定義を使用できるようにした。
Cosminexus サービスプラットフォームからレガシーシステムおよび ERP パッケージとの連携ができるようにした。
Cosminexus サービスプラットフォームの開発環境を一括構築できるようにした。
Cosminexus Component Container で使用できる機能として、Connector 1.5 を追加した。
セキュリティフィルタとセキュリティアナライザについての記述を削除した。
JP1 と連携した稼働性能の集中監視および稼働性能情報の収集と分析の機能を削除した。
JP1/NETM/Audit と連携した監査ログの収集に対応した。
アプリケーションの実行環境の構築の前提を、Smart Composer 機能を使用することに変更した。
実行環境で、データベースとして SQL Server 2005 を使用できるようにした。
実行環境で、XDM/RD E2 との連携に HiRDB Type4 JDBC Driver を使用できるようにした。
AIX 5L V5.1 のサポート中止に伴い、この OS に適応していた次の製品の記述を削除した。 <ul style="list-style-type: none"> • uCosminexus Application Server Enterprise • uCosminexus Application Server Standard • uCosminexus Service Platform • uCosminexus Web Redirector

追加・変更内容

Microsoft(R) Windows(R) 2000 Advanced Server Operating System , Microsoft(R) Windows(R) 2000 Datacenter Server Operating System , および Microsoft(R) Windows(R) 2000 Server Operating System のサポート中止に伴い、これらの OS に適応していた次の製品の記述を削除した。

- uCosminexus Application Server Enterprise
- uCosminexus Application Server Standard
- uCosminexus Client
- uCosminexus Developer Professional
- uCosminexus Developer Standard
- uCosminexus Operator
- uCosminexus Service Architect
- uCosminexus Service Platform
- uCosminexus Web Redirector

Microsoft(R) Windows(R) 2000 Professional Operating System のサポート中止に伴い、この OS に適応していた次の製品の記述を削除した。

- uCosminexus Client
- uCosminexus Developer Professional
- uCosminexus Developer Standard
- uCosminexus Operator
- uCosminexus Service Architect

次の製品の適用 OS に、HP-UX (IPF) を追加した。

- uCosminexus Service Platform

次の製品の適用 OS に、HP-UX 11i V3 (IPF) を追加した。

- uCosminexus Application Server Enterprise
- uCosminexus Application Server Standard
- uCosminexus Web Redirector

次の製品の適用 OS に、Red Hat Enterprise Linux ES 3 (AMD64 & Intel EM64T) , および Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) を追加した。

- uCosminexus Application Server Enterprise
- uCosminexus Application Server Standard
- uCosminexus Service Platform
- uCosminexus Web Redirector

次の製品の適用 OS に、Windows Vista を追加した。

- uCosminexus Client
 - uCosminexus Developer Professional
 - uCosminexus Developer Standard
 - uCosminexus Operator
 - uCosminexus Service Architect
-

変更内容 (3020-3-M01-20) uCosminexus Application Server Standard 07-10 , uCosminexus Application Server Enterprise 07-10 , uCosminexus Developer Standard 07-10 , uCosminexus Developer Professional 07-10 , uCosminexus Client 07-10 , uCosminexus Web Redirector 07-10 ,

uCosminexus Service Platform 07-10 , uCosminexus Service Architect 07-10 , uCosminexus Operator 07-10

追加・変更機能

uCosminexus Application Server Standard , uCosminexus Application Server Enterprise および uCosminexus Web Redirector の適用 OS に , HP-UX (PA-RISC) を追加した。

HiRDB Version 8 に対応した。また , uCosminexus Developer Standard , uCosminexus Developer Professional および uCosminexus Service Architect の構成ソフトウェアを , HiRDB Embedded Server Version 8 に変更した。

uCosminexus Application Server Standard , uCosminexus Application Server Enterprise および uCosminexus Web Redirector の適用 OS に , Linux (IPF) を追加した。

uCosminexus Service Platform の適用 OS に , AIX を追加した。

JP1 Version 8 に対応した。

Smart Composer 機能を使用した推奨構成として , CTM を使用したバックシステムを構築できるようにした。

Cosminexus サービスプラットフォームの実行環境を , 負荷分散を目的とした冗長構成で構築できるようにした。

はじめに

このマニュアルは、Cosminexus（コズミネクサス）の概要について説明したものです。Cosminexus の製品構成と、これらの製品で実現できることについて説明しています。また、Cosminexus を使用する場合に参照する必要があるマニュアルについても説明しています。Cosminexus では、次に示すプログラムプロダクトを使用してシステムの構築・運用、またはアプリケーションの開発をします。

- P-1J43-7D71 uCosminexus Application Server Standard
- P-1J43-7K71 uCosminexus Application Server Enterprise
- P-1J43-7M71 uCosminexus Web Redirector
- P-1M43-7D71 uCosminexus Application Server Standard
- P-1M43-7K71 uCosminexus Application Server Enterprise
- P-1M43-7M71 uCosminexus Web Redirector
- P-1M43-7S71 uCosminexus Service Platform
- P-2443-7D74 uCosminexus Application Server Standard
- P-2443-7E74 uCosminexus Developer Standard
- P-2443-7F74 uCosminexus Developer Professional
- P-2443-7H74 uCosminexus Client
- P-2443-7K74 uCosminexus Application Server Enterprise
- P-2443-7M74 uCosminexus Web Redirector
- P-2443-7S74 uCosminexus Service Platform
- P-2443-7T74 uCosminexus Service Architect
- P-2443-7U74 uCosminexus Operator
- P-9D43-7D71 uCosminexus Application Server Standard
- P-9D43-7K71 uCosminexus Application Server Enterprise
- P-9D43-7M71 uCosminexus Web Redirector
- P-9D43-7S71 uCosminexus Service Platform
- P-9S43-7D71 uCosminexus Application Server Standard
- P-9S43-7K71 uCosminexus Application Server Enterprise
- P-9S43-7M71 uCosminexus Web Redirector
- P-9S43-7S71 uCosminexus Service Platform
- P-9V43-7D71 uCosminexus Application Server Standard
- P-9V43-7K71 uCosminexus Application Server Enterprise
- P-9V43-7M71 uCosminexus Web Redirector

また、オペレーティングシステム（OS）の種類によって、機能が異なる場合があります。OS ごとの違いがある場合の表記方法については、「適用 OS の違いによる機能相違点の表記」を参照してください。

対象読者

このマニュアルは、Cosminexus を導入してシステムを設計・構築または運用する方、および Cosminexus で動作するアプリケーションを開発する方を対象としています。

次の内容を理解されていることを前提としています。

- OS (Windows または UNIX) に関する知識
- J2EE に関する知識

マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 Cosminexus の概要

Cosminexus の概要、および Cosminexus で構築できる環境について説明しています。

第 2 章 Cosminexus の製品構成

Cosminexus の製品構成について説明しています。

第 3 章 システムのライフサイクル

Cosminexus で実現するシステムのライフサイクル、およびライフサイクルを構成する段階について説明しています。

第 4 章 J2EE アプリケーションを実行する

Cosminexus で構築・運用できる J2EE アプリケーションの実行環境の概要について説明しています。

第 5 章 バッチアプリケーションを実行する

Cosminexus で構築・運用できるバッチアプリケーションの実行環境の概要について説明しています。

第 6 章 アプリケーションを開発する

Cosminexus の開発環境が提供する開発支援機能を使用したアプリケーションの開発について説明しています。

第 7 章 サービスの統合を実現する

Cosminexus で実現できるサービス統合について説明しています。

第 8 章 ほかの製品との連携

Cosminexus とほかの製品との連携について説明しています。

第 9 章 Cosminexus のマニュアル体系

Cosminexus の実行環境、開発環境およびサービス統合環境に対応するマニュアルの体系について説明しています。

付録 A Cosminexus が対応する標準仕様

Cosminexus が対応している標準仕様について説明しています。

付録 B 用語解説

このマニュアルで使用している用語の意味について説明しています。

関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。これらのマニュアルの概要、およびこれらのマニュアルを参照する目的については、「9. Cosminexus のマニュアル体系」を参照してください。

- Cosminexus 機能解説 (3020-3-M03)
- Cosminexus システム設計ガイド (3020-3-M04)
- Cosminexus 簡易構築・運用ガイド (3020-3-M05)
- Cosminexus システム構築ガイド (3020-3-M06)
- Cosminexus システム運用ガイド (3020-3-M07)
- Cosminexus アプリケーション設定操作ガイド (3020-3-M08)
- Cosminexus 運用管理操作ガイド (3020-3-M09)
- Cosminexus リファレンス コマンド編 (3020-3-M10)
- Cosminexus リファレンス 定義編 (3020-3-M11)
- Hitachi Web Server (3020-3-M15)
- TPBroker ユーザーズガイド (3020-3-M16)
- Cosminexus Reliable Messaging (3020-3-M17)
- Cosminexus メッセージ 1 KAWS / KDAL / KDJE 編 (3020-3-M20)
- Cosminexus メッセージ 2 KEOS / KEUC / KFCB 編 (3020-3-M21)
- Cosminexus メッセージ 3 KFCT / KFDB / KFDJ 編 (3020-3-M22)
- Cosminexus アプリケーション開発ガイド (3020-3-M41)
- Cosminexus リファレンス API 編 (3020-3-M42)
- Cosminexus XML Processor ユーザーズガイド (3020-3-M44)
- Borland(R) Enterprise Server VisiBroker(R) デベロッパーズガイド (3020-3-M45)
- Borland(R) Enterprise Server VisiBroker(R) プログラマーズリファレンス (3020-3-M46)
- Cosminexus SOAP アプリケーション開発ガイド (3020-3-M47)
- Cosminexus Web サービスセキュリティ 使用の手引 (3020-3-M48)
- Cosminexus XML Security - Core ユーザーズガイド (3020-3-M49)
- Cosminexus サービスプラットフォーム 概説 (3020-3-M71)
- Cosminexus サービスプラットフォーム 開発ガイド (3020-3-M72)
- Cosminexus サービスプラットフォーム システム構築・運用ガイド (3020-3-M73)
- Cosminexus Application Designer 操作ガイド (3020-3-M91)
- uCosminexus UDDI Registry システム運用ガイド (3020-3-J37)

このマニュアルでの表記

このマニュアルで使用している表記と、対応する製品名を次に示します。

表記		製品名
Application Server	Application Server Enterprise	uCosminexus Application Server Enterprise
	Application Server Standard	uCosminexus Application Server Standard
Developer	Developer Professional	uCosminexus Developer Professional
	Developer Standard	uCosminexus Developer Standard
HiRDB または HiRDB サーバ	HiRDB/Parallel Server	HiRDB/Parallel Server Version 7
		HiRDB/Parallel Server Version 8
	HiRDB/Single Server	HiRDB/Single Server Version 7
		HiRDB/Single Server Version 8
HiRDB Run Time または HiRDB クライアント		HiRDB/Run Time Version 7
		HiRDB/Run Time Version 8
Internet Explorer		Microsoft(R) Internet Explorer(R)
IPF		Itanium(R) Processor Family
JP1/AJS2	JP1/AJS2 - Agent	JP1/Automatic Job Management System 2 - Agent
	JP1/AJS2 - Light Edition	JP1/Automatic Job Management System 2 - Light Edition
	JP1/AJS2 - Manager	JP1/Automatic Job Management System 2 - Manager
	JP1/AJS2 - View	JP1/Automatic Job Management System 2 - View
JP1/AJS2 - SO	JP1/AJS2 - SO Manager	JP1/Automatic Job Management System 2 - Scenario Operation Manager
	JP1/AJS2 - SO View	JP1/Automatic Job Management System 2 - Scenario Operation View
JP1/Cm2	JP1/Cm2/ESA	JP1/Cm2/Extensible SNMP Agent
	JP1/Cm2/ESA for Extension Mib Runtime	JP1/Cm2/Extensible SNMP Agent for Extension Mib Runtime
JP1/Cm2/NC	JP1/Cm2/NC - Manager	JP1/Cm2/Network Element Configuration - Manager
	JP1/Cm2/NC - Manager for Cosminexus	JP1/Cm2/Network Element Configuration - Manager for Cosminexus
JP1/ESP		JP1/Extensible Service Probe

表記		製品名	
JP1/IM	JP1/IM - Central Console		JP1/Integrated Manager - Central Console
	JP1/IM - CM		JP1/Integrated Management - Central Information Master
			JP1/Integrated Manager - Central Information Master
	JP1/IM - Central Scope		JP1/Integrated Manager - Central Scope
	JP1/IM - Manager		JP1/Integrated Management - Manager
	JP1/IM - View		JP1/Integrated Management - View
JP1/Integrated Manager - View			
JP1/NETM/Audit		JP1/NETM/Audit - Manager	
JP1/PFM		JP1/Performance Management	
JP1/SC/DPM		JP1/ServerConductor/Deployment Manager Standard Edition	
JSF		JavaServer™ Faces Reference Implementation (RI) Version: 1.1_01 FCS	
Microsoft Cluster Service		Microsoft(R) Cluster Service	
Microsoft IIS	Microsoft IIS 6.0		Microsoft(R) Internet Information Services 6.0
Oracle	Oracle9i		Oracle9i
	Oracle9i R2		Oracle9i R2
	Oracle10g		Oracle 10g
	Oracle10g R2		Oracle 10gR2
Service Architect		uCosminexus Service Architect	
Service Platform		uCosminexus Service Platform	
SQL Server	SQL Server 2000		Microsoft(R) SQL Server(TM) 2000
	SQL Server 2005		Microsoft(R) SQL Server(TM) 2005
SQL Server の JDBC ドライバ	SQL Server 2000 Driver for JDBC		Microsoft(R) SQL Server(TM) 2000 Driver for JDBC
	SQL Server 2005 JDBC Driver		Microsoft(R) SQL Server(TM) 2005 JDBC Driver
Struts		Jakarta Struts 1.1	
UNIX	AIX		AIX 5L V5.2
			AIX 5L V5.3
	HP-UX	HP-UX (IPF)	HP-UX 11i V2 (IPF)
			HP-UX 11i V3 (IPF)
	Linux	Linux (IPF)	Red Hat Enterprise Linux AS 3 (IPF)
			Red Hat Enterprise Linux AS 4 (IPF)

表記		製品名	
		Red Hat Enterprise Linux 5 Advanced Platform (Intel Itanium)	
		Linux (x86 / AMD64 & Intel EM64T)	Red Hat Enterprise Linux AS 3 (x86)
			Red Hat Enterprise Linux AS 4 (x86)
			Red Hat Enterprise Linux 5 Advanced Platform (x86)
			Red Hat Enterprise Linux ES 3 (x86)
			Red Hat Enterprise Linux ES 4 (x86)
			Red Hat Enterprise Linux 5 (x86)
			Red Hat Enterprise Linux AS 3 (AMD64 & Intel EM64T)
			Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T)
			Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64)
			Red Hat Enterprise Linux ES 3 (AMD64 & Intel EM64T)
			Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T)
			Red Hat Enterprise Linux 5 (AMD/Intel 64)
		Solaris	Solaris 9
Solaris 10			
Windows Server 2003		Microsoft(R) Windows Server(R) 2003 , Enterprise Edition Operating System (x86)	
		Microsoft(R) Windows Server(R) 2003 , Standard Edition Operating System (x86)	
Windows Server 2003 R2		Microsoft(R) Windows Server(R) 2003 R2 , Enterprise Edition Operating System (x86)	
		Microsoft(R) Windows Server(R) 2003 R2 , Standard Edition Operating System (x86)	
Windows Server 2003 (x64)		Microsoft(R) Windows Server(R) 2003 , Enterprise x64 Edition Operating System	
		Microsoft(R) Windows Server(R) 2003 , Standard x64 Edition Operating System	

表記		製品名
Windows Server 2003 R2 (x64)		Microsoft(R) Windows Server(R) 2003 R2 , Enterprise x64 Edition Operating System
		Microsoft(R) Windows Server(R) 2003 R2 , Standard x64 Edition Operating System
Windows Vista	Windows Vista Business	Microsoft(R) Windows Vista(R) Business
	Windows Vista Enterprise	Microsoft(R) Windows Vista(R) Enterprise
	Windows Vista Ultimate	Microsoft(R) Windows Vista(R) Ultimate
Windows XP		Microsoft(R) Windows(R) XP Professional Operating System
XDM/RD E2		VOS3 XDM/RD E2

なお，Windows Server 2003，Windows Server 2003 R2，Windows Server 2003 (x64)，Windows Server 2003 R2 (x64)，Windows Vista および Windows XP を総称して Windows と表記することがあります。

このマニュアルで使用している表記と，対応する Cosminexus の機能名を次に示します。

表記	Cosminexus の機能名
Cosminexus RM	Cosminexus Reliable Messaging
Cosminexus Developer's Kit for Java	Cosminexus Developer's Kit for Java™
CTM	Cosminexus Component Transaction Monitor
DB Connector for Cosminexus RM	DB Connector for Cosminexus Reliable Messaging
Management Server	Cosminexus Management Server
MyEclipse	MyEclipse for Cosminexus
PRF	Cosminexus Performance Tracer
Server Plug-in	Cosminexus Server Plug-in
Smart Composer	Cosminexus Smart Composer
TPBroker	Cosminexus TPBroker

このマニュアルで使用している表記と，対応する Java 関連用語を次に示します。

表記	Java 関連用語
Connector 1.0	J2EE™ Connector Architecture 1.0
Connector 1.5	J2EE™ Connector Architecture 1.5
DI	Dependency Injection
EAR	Enterprise ARchive
EJB または Enterprise JavaBeans	Enterprise JavaBeans™

表記	Java 関連用語
EJB QL	EJB TM Query Language
J2EE または Java 2 Platform, Enterprise Edition	Java TM 2 Platform, Enterprise Edition
J2SE	Java TM 2 Platform, Standard Edition
JAAS	Java TM Authentication and Authorization Service
JAR	Java TM Archive
Java	Java TM
Java 2 Runtime Environment, Standard Edition	Java TM 2 Runtime Environment, Standard Edition
Java 2 SDK, Standard Edition	Java TM 2 Software Development Kit, Standard Edition
JavaBeans	JavaBeans TM
JavaMail	JavaMail TM
JAXP	Java TM API for XML Processing
JCA	J2EE TM Connector Architecture
JCE	Java TM Cryptography Extension
JDBC	JDBC TM
	Java TM Database Connectivity
JDK	Java TM Development Kit
JMS	Java TM Message Service
JNDI	Java TM Naming and Directory Interface TM
JNI	Java TM Native Interface
JSF	JavaServer TM Faces Reference Implementation (RI) Version: 1.1_01 FCS
JSP	JavaServer Pages TM
JTA	Java TM Transaction API
JTS	Java TM Transaction Service
Servlet またはサーブレット	Java TM Servlet
WAR	Web ARchive

適用 OS の違いによる機能相違点の表記

このマニュアルは、適用 OS が Windows , AIX , HP-UX , Linux , および Solaris の製品に対応します。OS によって記述を書き分ける場合、次に示す表記を使用して、それぞれの説明に OS 名を明記しています。

表記	意味
Windows の場合	Windows に該当する表記です。
AIX の場合	AIX に該当する表記です。
HP-UX の場合	HP-UX に該当する表記です。
Linux の場合	Linux に該当する表記です。
Solaris の場合	Solaris に該当する表記です。
UNIX の場合	UNIX (AIX , HP-UX , Linux , Solaris) に該当する表記です。

このマニュアルで使用している略語

このマニュアルで使用している英略語を次に示します。

英略語	英字での表記
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
ASCII	<u>A</u> merican <u>S</u> tandard <u>C</u> ode for <u>I</u> nformation <u>I</u> nterchange
BMP	<u>B</u> ean- <u>M</u> anaged <u>P</u> ersistence
BMT	<u>B</u> ean- <u>M</u> anaged <u>T</u> ransaction
BPEL	<u>B</u> usiness <u>P</u> rocess <u>E</u> xecution <u>L</u> anguage
CMP	<u>C</u> ontainer- <u>M</u> anaged <u>P</u> ersistence
CMR	<u>C</u> ontainer- <u>M</u> anaged <u>R</u> elationship
CMT	<u>C</u> ontainer- <u>M</u> anaged <u>T</u> ransaction
CORBA	<u>C</u> ommon <u>O</u> bject <u>R</u> equest <u>B</u> roker <u>A</u> rchitecture
CSV	<u>C</u> omma <u>S</u> eparated <u>V</u> alue
CUI	<u>C</u> haracter <u>U</u> ser <u>I</u> nterface
DB	<u>D</u> atabase
DBMS	<u>D</u> atabase <u>M</u> anagement <u>S</u> ystem
DD	<u>D</u> eployment <u>D</u> escriptor
DIT	<u>D</u> irectory <u>I</u> nformation <u>T</u> ree
DMZ	<u>D</u> emilitarized <u>Z</u> one
DN	<u>D</u> istinguished <u>N</u> ame
DNS	<u>D</u> omain <u>N</u> ame <u>S</u> ystem
DTD	<u>D</u> ocument <u>T</u> ype <u>D</u> efinition
EIS	<u>E</u> nterprise <u>I</u> nformation <u>S</u> ystem
EL	<u>E</u> xpression <u>L</u> anguage
ERP	<u>E</u> nterprise <u>R</u> esource <u>P</u> lanning

英略語	英字での表記
ESB	<u>E</u> nterprise <u>S</u> ervice <u>B</u> us
EUC	<u>E</u> xtended <u>U</u> NIX <u>C</u> ode
GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
HA	<u>H</u> igh <u>A</u> vailability
HTML	<u>H</u> yper <u>T</u> ext <u>M</u> arkup <u>L</u> anguage
HTTP	<u>H</u> yper <u>T</u> ext <u>T</u> ransfer <u>P</u> rotocol
HTTPS	<u>H</u> yper <u>T</u> ext <u>T</u> ransfer <u>P</u> rotocol <u>S</u> ecurity
IDE	<u>I</u> ntegrated <u>D</u> evelopment <u>E</u> nvironment
IOP	<u>I</u> nternet <u>I</u> nter- <u>O</u> rb <u>P</u> rotocol
JIS	<u>J</u> apanese <u>I</u> ndustrial <u>S</u> tandards
LAN	<u>L</u> ocal <u>A</u> rea <u>N</u> etwork
LDAP	<u>L</u> ightweight <u>D</u> irectory <u>A</u> ccess <u>P</u> rotocol
LDIF	<u>L</u> DAP <u>D</u> ata <u>I</u> nterchange <u>F</u> ormat
MDA	<u>M</u> odel <u>D</u> riven <u>A</u> rchitecture
MIB	<u>M</u> anagement <u>I</u> nformation <u>B</u> ase
OID	<u>O</u> bject <u>I</u> dentifier
OLTP	<u>O</u> n- <u>L</u> ine <u>T</u> ransaction <u>P</u> rocessing
OMG	<u>O</u> bject <u>M</u> anagement <u>G</u> roup
ORB	<u>O</u> bject <u>R</u> equest <u>B</u> roker
OS	<u>O</u> perating <u>S</u> ystem
OTM	<u>O</u> bject <u>T</u> ransaction <u>M</u> onitor
OTS	<u>O</u> bject <u>T</u> ransaction <u>S</u> ervice
PIM	<u>P</u> latform <u>I</u> ndependent <u>M</u> odel
POA	<u>P</u> ortable <u>O</u> bject <u>A</u> dapter
PSM	<u>P</u> latform <u>S</u> pecific <u>M</u> odel
RAC	<u>R</u> eal <u>A</u> pplication <u>C</u> lusters
RDB	<u>R</u> elational <u>D</u> atabase
RMI	<u>R</u> emote <u>M</u> ethod <u>I</u> nvocation
RPC	<u>R</u> emote <u>P</u> rocedure <u>C</u> all
SFO	<u>S</u> ession <u>F</u> ail <u>O</u> ver
SHA	<u>S</u> ecure <u>H</u> ash <u>A</u> lgorithm
SNMP	<u>S</u> imple <u>N</u> etwork <u>M</u> anagement <u>P</u> rotocol
SOA	<u>S</u> ervice <u>O</u> riented <u>A</u> rchitecture

英略語	英字での表記
SOAP	<u>S</u> imple <u>O</u> bject <u>A</u> ccess <u>P</u> rotocol
SPI	<u>S</u> ervice <u>P</u> rovider <u>I</u> nterface
SPP	<u>S</u> ervice <u>P</u> roviding <u>P</u> rogram
SSL	<u>S</u> ecure <u>S</u> ockets <u>L</u> ayer
TCS	<u>T</u> ransaction <u>C</u> ontext <u>S</u> erver
UDDI	<u>U</u> niversal <u>D</u> escription, <u>D</u> iscovery and <u>I</u> ntegration
UML	<u>U</u> nified <u>M</u> odeling <u>L</u> anguage
URI	<u>U</u> niform <u>R</u> esource <u>I</u> dentifier
URL	<u>U</u> niform <u>R</u> esource <u>L</u> ocator
UTF	<u>U</u> CS <u>T</u> ransformation <u>F</u> ormat
VM	<u>V</u> irtual <u>M</u> achine
WSDL	<u>W</u> eb <u>S</u> ervice <u>D</u> escription <u>L</u> anguage
XML	<u>E</u> xtensible <u>M</u> arkup <u>L</u> anguage

このマニュアルの図中で使用している記号

このマニュアルの図中で使用している記号を、次のように定義します。

●サーバホスト



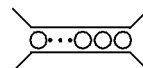
●クライアントホスト



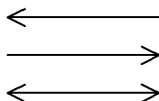
●データベース



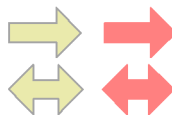
●キュー



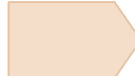
●制御の流れ



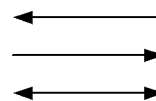
●データの流れ



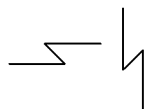
●工程、作業項目の流れ



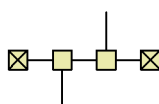
●その他の流れ



●通信回線



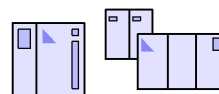
●バス型のLAN



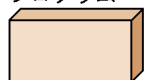
●ネットワーク



●Cosmi nexus以外のシステム



●アプリケーション、プログラム



このマニュアルで使用している記号

! 注意事項

間違いやすい点，および間違うとトラブルが発生するような項目について説明しています。

ポイント

その説明の要点について説明しています。

参考

補足的な情報を説明しています。

常用漢字以外の漢字の使用について

このマニュアルでは，常用漢字を使用することを基本としていますが，次に示す用語については，常用漢字以外の漢字を使用しています。

鍵（かぎ） 個所（かしよ） 必須（ひっす） 閉塞（へいそく） 漏洩（ろうえい）

KB（キロバイト）などの単位表記について

1KB（キロバイト），1MB（メガバイト），1GB（ギガバイト），1TB（テラバイト）はそれぞれ 1,024 バイト，1,024² バイト，1,024³ バイト，1,024⁴ バイトです。

目次

1	Cosminexus の概要	1
1.1	Cosminexus とは	2
1.2	Cosminexus の特長	3
1.2.1	Cosminexus で実現できること	3
1.2.2	J2EE アプリケーション実行環境の概要と特長	5
1.2.3	バッチアプリケーション実行環境の概要と特長	9
1.2.4	アプリケーション開発環境の概要と特長	10
1.2.5	サービス統合環境の概要と特長	10
1.3	Cosminexus で構築できる環境	13
1.3.1	Cosminexus の実行環境を組み合わせたシステムの例	13
1.3.2	アプリケーションの実行環境と開発環境の関係	15
1.3.3	サービス統合環境の開発環境，運用環境および実行環境の関係	16
2	Cosminexus の製品構成	19
2.1	製品の分類と特長	20
2.1.1	実行環境を構築する製品	20
2.1.2	開発環境を構築する製品	21
2.1.3	サービス統合環境を構築する製品	21
2.1.4	オプション製品	22
2.2	製品の概要	23
2.2.1	Application Server Standard	23
2.2.2	Application Server Enterprise	24
2.2.3	Developer Standard	27
2.2.4	Developer Professional	28
2.2.5	Service Platform	28
2.2.6	Service Architect	30
2.2.7	uCosminexus Client	30
2.2.8	uCosminexus Web Redirector	31
2.2.9	uCosminexus Operator	32
2.3	構成ソフトウェア	34
2.3.1	製品と構成ソフトウェアの対応	34
2.3.2	構成ソフトウェアの機能概要	36
2.3.3	実行環境の構成ソフトウェアの位置づけ (Application Server)	41

2.3.4	開発環境の構成ソフトウェアの位置づけ (Developer)	46
2.3.5	サービス統合環境の構成ソフトウェアの位置づけ (Service Platform , Service Architect)	48
2.4	動作環境	50

3

	システムのライフサイクル	53
3.1	実行環境および開発環境として使用する場合のシステムのライフサイクル	54
3.1.1	システムの仕様 (機能) 検討	56
3.1.2	システム設計と運用設計	60
3.1.3	アプリケーションの開発	61
3.1.4	システムの構築	62
3.1.5	システムの運用と保守	62
3.2	サービス統合環境として使用する場合のライフサイクル	64
3.2.1	サービスの準備	64
3.2.2	各種定義 / アプリケーションの作成	64

4

	J2EE アプリケーションを実行する	65
4.1	J2EE アプリケーションの実行環境の特長	66
4.1.1	標準仕様への対応	66
4.1.2	システムの安定稼働の実現	66
4.1.3	可用性と耐障害性の向上	69
4.1.4	システム導入および拡張の容易化	71
4.1.5	システム監査によるシステムのセキュリティ確保	72
4.1.6	業務効率を向上させる運用管理の実現	75
4.1.7	Web サービスへの対応	77
4.1.8	信頼性の高い非同期通信の実現	78
4.2	J2EE アプリケーションの実行環境の種類	79
4.3	J2EE アプリケーションの実行環境を構築・運用する	80
4.3.1	システム構成の種類	80
4.3.2	スケールアウト構成のシステムの構築・運用	82
4.3.3	フロントエンド / バックエンド構成のシステムの構築・運用	83
4.4	Web サービスに対応したアプリケーションの実行環境を構築・運用する	85
4.4.1	SOAP アプリケーションの実行環境	85
4.4.2	Web サービスセキュリティの技術を利用した SOAP アプリケーションの実行環境	86

4.5	メッセージを利用した非同期通信アプリケーションの実行環境を構築・運用する	88
4.6	導入から運用までの流れ	91

5

	バッチアプリケーションを実行する	95
5.1	バッチアプリケーションの実行環境の特長	96
5.1.1	オープン環境でのバッチジョブの実行	96
5.1.2	コネクションプール/ステートメントプールを使用したデータベースアクセスの高速化	97
5.1.3	リソース排他状態でのフルガーベージコレクション実行を抑止	97
5.1.4	そのほかの特長	99
5.2	バッチアプリケーションの実行環境を構築・運用する	100
5.2.1	システム構成の種類	100
5.2.2	バッチアプリケーションの実行環境の構築・運用	101
5.3	導入から運用までの流れ	103

6

	アプリケーションを開発する	107
6.1	Cosminexus でのアプリケーション開発の特長	108
6.2	開発するアプリケーションの種類	110
6.3	J2EE アプリケーションを開発する	111
6.3.1	Developer が提供する MyEclipse およびプラグインを使用する	111
6.3.2	そのほかに使用できる機能	113
6.4	Web サービスに対応したアプリケーションを開発する	114
6.4.1	SOAP アプリケーションの開発支援機能を使用する	114
6.4.2	Web サービスセキュリティ技術を使用する	115
6.5	メッセージを使用した非同期通信用のアプリケーションを開発する	117
6.6	アプリケーション開発の流れ	118

7

	サービスの統合を実現する	123
7.1	サービス統合環境の特長	124
7.1.1	SOA の概念に基づくサービスインテグレーションの実現	124
7.1.2	レガシーシステムや ERP パッケージの統合	126
7.1.3	サービスコンポーネント開発の効率化	128
7.2	メッセージフローによってサービスを統合する	130
7.2.1	Cosminexus サービスプラットフォームを構成する三つの環境	130

7.2.2	Cosminexus サービスプラットフォームのシステム構成	133
7.2.3	サービスを利用する場合のリクエストの流れ	135
7.2.4	Cosminexus サービスプラットフォームの構築と運用	136
7.3	メッセージフローによるサービス統合に必要なコンポーネントを定義 / 開発する	137
7.3.1	サービスコンポーネントを定義する	137
7.3.2	サービスリクエストを開発する	137

8

ほかの製品との連携		139
8.1	データベースとの連携	140
8.1.1	J2EE サーバから接続できるデータベース	140
8.1.2	Cosminexus サービスプラットフォームで接続できるデータベース	142
8.2	JP1 との連携	144
8.3	クラスタソフトウェアとの連携	145

9

Cosminexus のマニュアル体系		147
9.1	マニュアル体系	148
9.2	マニュアルの読書手順	152
9.2.1	実行環境を構築・運用する場合の読書手順	152
9.2.2	アプリケーションを開発する場合の読書手順	155
9.2.3	サービス統合環境を構築・運用する場合のマニュアルの読書手順	156

付録

付録 A Cosminexus が対応する標準仕様		159
付録 B 用語解説		160

索引

169

1

Cosminexus の概要

この章では、Cosminexus の概要、および Cosminexus で構築できる環境ごとの特長について説明します。

Cosminexus (コズミネクサス) は、業務システムの中核に位置し、アプリケーションを実行する基盤となる製品です。標準技術である J2EE 1.4 に準拠したアプリケーションサーバを構築・運用できます。

また、アプリケーションサーバの機能を基盤として、SOA の概念を実現する環境を構築・運用することもできます。

さらに、アプリケーションサーバ上で実行するアプリケーションを効率良く開発する環境も構築できます。

1.1 Cosminexus とは

1.2 Cosminexus の特長

1.3 Cosminexus で構築できる環境

1.1 Cosminexus とは

情報社会が急速に変化している現在、ビジネスを成長させるためには市場ニーズに柔軟に対応できるインフラが必要です。このようなインフラを実現するためには、拡張性に優れた情報システムの構築が必須であり、情報システムの優劣はビジネスの成功を左右する程の影響力を持ちます。さらに、社会変化のスピードに合わせて業務のライフサイクルが短くなっていくのに伴って、業務の変化に柔軟に対応でき、かつ既存資産をむだなく再利用できることも、情報システムの必須要件になってきています。

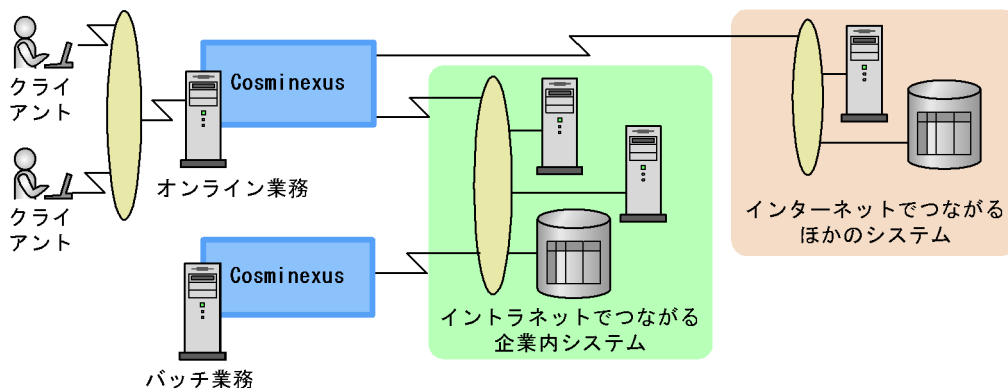
このような要件にこたえるシステムを実現するのが、Cosminexus です。

Cosminexus は、ビジネス環境やビジネス戦略の急激な変化に柔軟に対応できる、信頼性と拡張性の高い情報システムの基盤となる製品です。標準技術である J2EE に準拠したアプリケーションの実行基盤を構築できるので、J2EE の特長である柔軟性、信頼性の高いシステムを効率的に開発・運用できるようになります。

また、Cosminexus は、インターネットやイントラネットに存在するさまざまな既存資産を組み合わせて新しい価値を持つ資産として活用するための仕組みを持っています。これによって、インターネットで接続されたほかのシステムや、企業システム内に存在する既存資産を有効に活用しながら、新たなニーズに速やかに対応するための基盤になります。

Cosminexus の概要を次の図に示します。

図 1-1 Cosminexus の概要



Cosminexus は、オンライン業務とバッチ業務のそれぞれに適したシステムの基盤になります。

また、Cosminexus は、Java のほか、CORBA などの業界標準に準拠しています。このため、ほかのシステムとの連携や、ほかのシステムからの移植も円滑に実現できます。

1.2 Cosminexus の特長

この節では、Cosminexus の特長について説明します。

1.2.1 Cosminexus で実現できること

ここでは、Cosminexus で実現できることについて説明します。

Cosminexus の製品は、サービスを実行するための基盤となる製品（サービスデプロイメント）と、サービス統合の中核となるシステムを構築するための製品（サービスインテグレーション）で構成されています。

サービスデプロイメントおよびサービスインテグレーションに対応する製品を使用すると、次の 3 種類の環境を構築・運用できます。

アプリケーション実行環境

アプリケーションを実行することでユーザにサービスを提供する基盤となる環境です。

アプリケーション実行環境では、次の 2 種類の業務を実行できます。

- オンライン業務（オンライン処理）
インターネットやイントラネット上のユーザから送信された要求を随時処理する形式の業務です。オンライン業務では、J2EE の技術を使用して開発されたアプリケーションを実行します。この環境を、J2EE アプリケーション実行環境といいます。
- バッチ業務（バッチ処理）
定型的な業務を決まった時間にまとめて処理する形式の業務です。従来メインフレーム上などで実行されていたバッチジョブを、オープン環境の技術である Java を使用して実現できます。Java で開発したバッチジョブを実行するためのアプリケーションを、バッチアプリケーションといいます。バッチアプリケーションを実行するための環境を、バッチアプリケーション実行環境といいます。

アプリケーション開発環境

アプリケーション実行環境上で動作するアプリケーションを開発する環境です。Cosminexus では、実行環境上で動作するアプリケーションの設計・開発からデバッグまでを統括的に支援する開発環境を構築・運用できます。

サービス統合環境

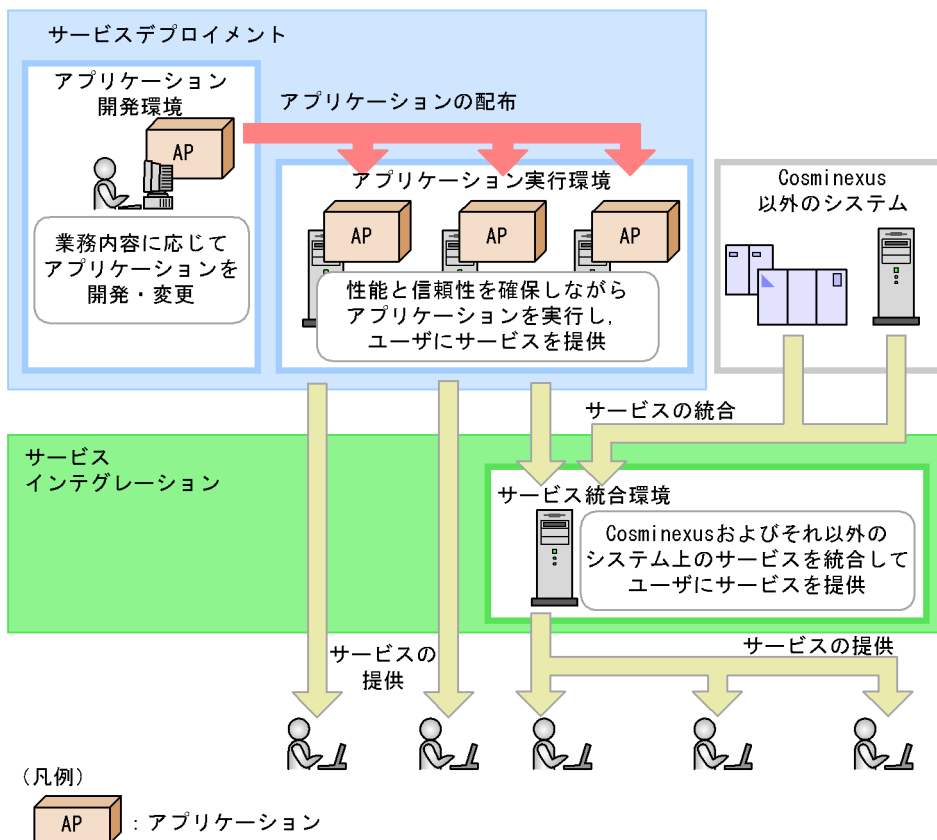
J2EE アプリケーション実行環境で稼働しているサービスや、Cosminexus 以外の環境で稼働しているサービスを統合して、新しいサービスとしてユーザに提供するための基盤となる環境です。Cosminexus では、サービスの組み替えや新しいサービスの追加などを柔軟に実行できるサービス統合環境を構築・運用できます。

Cosminexus では、これらの環境の組み合わせによって、多様な要件の業務システムの構築を実現します。

1. Cosminexus の概要

3種類の環境の概要を次の図に示します。

図 1-2 3種類の環境の概要



アプリケーション実行環境では、性能と信頼性を確保しながら、ユーザからのリクエストを迅速に処理して、ユーザにサービスを提供します。また、バッチジョブを効率良く実行します。可用性や耐障害性の高い実行基盤として、24時間連続稼働が必要なサービス、停止することが許されない重要な業務、バッチジョブとして実行される基幹業務など、さまざまな要件のアプリケーションを実行するための環境になります。また、セキュリティに配慮したシステムや、内部統制に対応したシステムを実現する基盤としての機能も備えています。

オンライン処理に対応する J2EE アプリケーション実行環境の概要と特長については、「1.2.2 J2EE アプリケーション実行環境の概要と特長」で説明します。バッチ処理に対応するバッチアプリケーション実行環境の概要と特長については、「1.2.3 バッチアプリケーション実行環境の概要と特長」で説明します。

業務内容の変更や組織の組み替えなどが発生した場合に、新しいアプリケーションを開発したり、既存のアプリケーションを変更したりするためには、アプリケーション開発環境を使用できます。アプリケーション開発環境では、画面設計から、バックシステム

と連携する業務処理の実装まで、シームレスに効率良く実行できます。さらに、開発したアプリケーションを、実行環境と同じ機能を持つテスト環境でテストすることもできます。アプリケーション開発環境の概要と特長については、「1.2.4 アプリケーション開発環境の概要と特長」で説明します。

業務内容の変更や組織の組み替えなどに対して、アプリケーションの開発で対応するのではなく、既存のアプリケーションや他システムが提供するサービスの再利用で対応する方法もあります。サービス統合環境では、アプリケーション実行環境が提供するサービスや Cosminexus 以外が提供するサービスを統合して、新しいサービスとしてユーザに提供できます。また、サービス統合環境は、複数のアプリケーションやシステムから提供される既存のサービスを、統一されたインターフェースで提供するための基盤としても使用できます。サービス統合環境の概要と特長については、「1.2.5 サービス統合環境の概要と特長」で説明します。

1.2.2 J2EE アプリケーション実行環境の概要と特長

ここでは、J2EE アプリケーション実行環境の概要と特長について説明します。詳細については、「4.1 J2EE アプリケーションの実行環境の特長」を参照してください。

(1) アプリケーション実行環境の概要

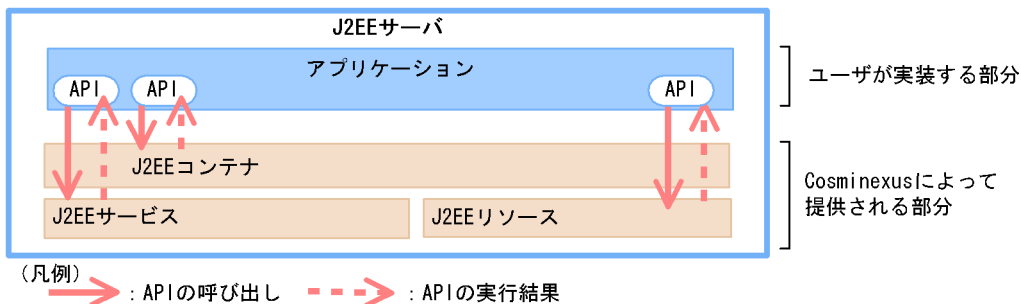
Cosminexus は、標準仕様である J2EE に準拠したアプリケーションの実行環境を構築します。J2EE に準拠したアプリケーションを実行する機能を持つサーバプロセスを、J2EE サーバといいます。

J2EE サーバは、J2EE コンテナ、J2EE サービス、J2EE リソースなどの J2EE で規定された仕様に従って、ユーザが開発した J2EE アプリケーションを実行するために必要な機能を提供します。例えば、J2EE サーバでは、トランザクション管理やセキュリティ管理など、複数の業務に共通する処理を実行する機能を、J2EE サービスや J2EE リソースによって提供しています。アプリケーション開発時には、アプリケーション内で J2EE コンテナ、J2EE サービス、J2EE リソースなどで提供されている API を呼び出すことで、煩雑なコーディングをしないで、複数の業務で共通の処理を実現できます。

アプリケーションと J2EE サーバの関係を次の図に示します。

1. Cosminexus の概要

図 1-3 アプリケーションと J2EE サーバの関係



J2EE サーバを中心とした、アプリケーションの実行環境になるサーバ基盤を、アプリケーションサーバといいます。Cosminexus の実行環境は、アプリケーションサーバを中心に構成されます。

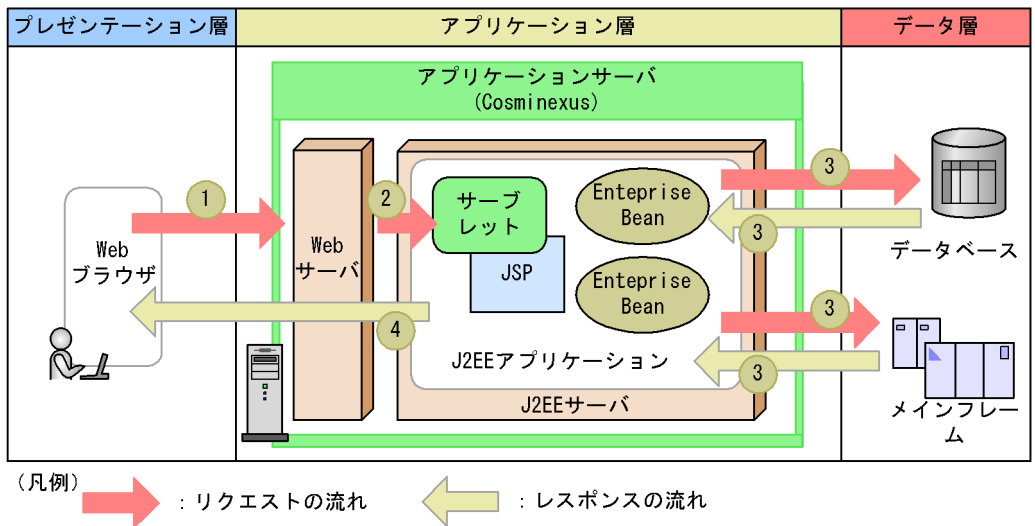
アプリケーションサーバは、情報システムの間中に位置し、ユーザの要求とデータベースなど、業務システム間の処理の受け渡しをするミドルウェアです。

アプリケーションサーバ上では、ユーザの要求に応じたサービスを提供するための J2EE アプリケーションが実行されます。J2EE アプリケーションとは、J2EE サーバ上で動作する、実現したい業務内容に応じて開発されたアプリケーションです。

J2EE アプリケーションは、ユーザからの要求を受け付け、処理を実行し、結果をユーザに返します。また、処理内容によっては、データベースやメインフレームなどのほかのシステムとデータをやり取りして、必要な情報を取得して処理を実行します。

アプリケーションサーバを中心としたシステムでのリクエスト処理の流れを次の図に示します。

図 1-4 アプリケーションサーバを中心としたシステムでのリクエスト処理の流れ



図で示した流れについて説明します。

1. ユーザが Web ブラウザ上で処理を実行すると、Web ブラウザからアプリケーションサーバに対してリクエストが送信されます。Web ブラウザからのリクエストは、アプリケーションサーバの一部である Web サーバが受け付けます。
2. アプリケーションサーバ内で、ユーザのリクエストに対応する J2EE アプリケーションが実行されます。J2EE アプリケーションは、Web サーバから転送されたリクエストを受け付けるためのプログラム（サブレット、JSP）や、業務処理を実行するプログラム（Enterprise Bean など）で構成されています。
3. J2EE アプリケーションは、必要に応じて、ユーザのリクエストを処理するためにデータベースやほかのシステムにアクセスします。
4. 業務処理が完了したら、Web サーバ経由で Web ブラウザにレスポンスが送信されます。Web ブラウザ上で表示するための画面は、J2EE アプリケーション内のサブレットや JSP で生成されます。生成された内容が、処理結果としてユーザが操作している Web ブラウザ上に表示されます。

Cosminexus では、安定稼働性や耐障害性が高く、優れたパフォーマンスを実現するシステムを構築できます。また、作業を円滑に実行するための機能によって、効率の良いシステム構築・システム運用を実現できます。

(2) J2EE アプリケーション実行環境の特長

J2EE アプリケーション実行環境の特長について説明します。

標準仕様への対応

J2EE 1.4 に準拠した J2EE サーバなど、標準仕様に準拠したアプリケーションサーバの構築・運用を実現できます。

1. Cosminexus の概要

システムの安定稼働の実現

Web アプリケーションに対する流量制御や、Enterprise Bean に対する流量制御、優先制御および負荷分散を実現できます。これによって、リクエスト数の急激な変動や一部のアプリケーションの高負荷状態にも柔軟に対応し、システムの安定稼働を実現できます。

可用性と耐障害性の向上

システムの稼働状態を監視して障害の予兆を検知し、障害を未然に防止したり、障害が発生した場合に速やかに自動回復したりする機能を備えています。これによって、可用性と耐障害性の高いシステムを実現できます。

システム導入および拡張の容易化

システムで設定が必要なパラメータを一括設定できます。システムのスケールアウト、スケールインにもスムーズに対応できます。

また、本番運用をする環境とは別のテスト環境でテストやチューニングを実施した場合に、チューニング済みの定義をまとめてファイルに出力できます。これを利用して本番環境を構築することで、テスト環境から本番環境への、構築済みのシステムのスムーズな移行を実現できます。

さらに、UNIX 上で Cosminexus を使用している場合、稼働中の実行環境の情報を退避して、同じマシンに別の実行環境を構築できます。この機能は、本番稼働中の実行環境をバージョンアップしたり、本番稼働中の実行環境にパッチを適用したりする場合に有効です。バージョンアップやパッチ適用後に問題が発生した場合などに、速やかに元の環境を復元できます。

システム監査によるシステムのセキュリティ確保

組織の内部統制強化のニーズなどにこたえるため、「だれが」「いつ」「何を実施したか」という情報をログとして残しておけます。このログを使用すると、システムが正しい運用手順で管理・運用されていることを監査・証明できます。

また、データベースの機能と連携して、データベースアクセスの監査に有効な情報も出力できます。出力された情報を基に、Cosminexus を経由して実行されたデータベースへのアクセスが、どのクライアントが送信したリクエストで実行されたのかを判別できます。これによって、データベースアクセスを伴う操作が、だれによって実施されたのかを監査・証明できます。

業務効率を向上させる運用管理の実現

複数のマシンで構成されるシステムを一つのマシンから一括運用できます。また、ほかの運用管理プログラムと連携することで、Cosminexus 以外のシステムも含めた自動運用を実現できます。

Web サービスへの対応

Web サービスに対応した実行環境を構築できます。また、Web サービスを利用するための SOAP アプリケーション開発を支援する機能も使用できます。

信頼性の高い非同期通信の実現

メッセージを使用した非同期通信を実現するための実行環境を構築し、送信側と受信側での処理の非同期実行を実現できます。また、メッセージをデータベース上の

キューで管理することで、信頼性の高い非同期通信を実現できます。なお、メッセージの実装には、Cosminexus が提供する JMS インタフェースを利用できます。

1.2.3 バッチアプリケーション実行環境の概要と特長

ここでは、バッチアプリケーション実行環境の概要と特長について説明します。詳細については、「5.1 バッチアプリケーションの実行環境の特長」を参照してください。

(1) バッチアプリケーション実行環境の概要

バッチアプリケーションの実行環境では、Java プログラムとして開発されたバッチアプリケーションを実行します。従来メインフレーム上で動作していたバッチジョブのオープン環境への移行を図れます。

バッチアプリケーションはサーバプロセス上で動作します。バッチアプリケーションを実行する機能を持つサーバプロセスを、バッチサーバといいます。バッチサーバを使用することによって、アプリケーションの実行ごとに JavaVM を起動するコストを抑えられます。これによって、効率良くバッチジョブを実行できます。

(2) バッチアプリケーション実行環境の特長

バッチアプリケーション実行環境の特長について説明します。

オープン環境でのバッチジョブの実行

Java の特長である柔軟性や運用容易性を兼ね備えたバッチジョブの実行環境を構築・運用できます。また、JP1/AJS2 や uCosminexus Batch Job Execution Server などの関連プログラムと連携することで、ジョブの自動実行や、メインフレームで実現していた内容に類似した処理の制御を実現できます。

コネクションプール/ステートメントプールを使用したデータベースアクセスの高速化

バッチアプリケーション実行環境では、コネクションプールやステートメントプールなどの機能を使用できます。これによって、バッチアプリケーションからデータベースへのアクセスを高速にできます。なお、コネクションプールおよびステートメントプールの機能は、リソースアダプタを使用して実現します。

リソース排他状態でのフルガーベージコレクション実行を抑制

バッチアプリケーション実行環境では、バッチサーバ上でフルガーベージコレクションを実行するタイミングを制御できます。この制御では、リソースを排他していないタイミングを見計らってフルガーベージコレクションを実行して、不要なオブジェクトを解放します。これによって、意図しないタイミングでフルガーベージコレクションが実行されることを抑制できます。

この機能は、バッチアプリケーションとオンラインアプリケーションが同じリソースにアクセスする場合に有効です。

1.2.4 アプリケーション開発環境の概要と特長

ここでは、Cosminexus のアプリケーション開発環境の概要と特長について説明します。詳細については、「6.1 Cosminexus でのアプリケーション開発の特長」を参照してください。

(1) アプリケーション開発環境の概要

Cosminexus では、アプリケーションサーバ上で実行する J2EE アプリケーションを開発するために、J2EE アプリケーションの開発からデバッグまでを一連の流れで実行できる、統合開発環境を提供しています。Cosminexus の統合開発環境では、業務指向スタイルでの J2EE アプリケーションの開発を支援しています。業務指向スタイルとは、業務の流れに着目し、その業務を遂行するために必要な操作や手順を実装していく開発スタイルです。これによって、ユーザインタフェースの早期確定や、MVC アーキテクチャに従った効率の良い J2EE アプリケーション開発ができます。Cosminexus の開発環境では、プログラミング・デバッグから Cosminexus 独自の開発支援機能を使用したアプリケーションの実装までを、一貫したインタフェースで実行できます。

また、Web サービスに対応するための SOAP アプリケーションの開発も支援しています。SOAP アプリケーションは、Cosminexus で提供されている SOAP アプリケーション開発支援機能を使用して開発できます。また、SOAP アプリケーションで送受信する SOAP メッセージは、セキュリティ機能によって機密保護ができます。

さらに、Cosminexus では、開発環境の中に、テスト用の実行環境を構築できます。このため、アプリケーションの設計・プログラミングからテスト・デバッグまで、アプリケーション開発に必要な一連の作業を、すべて開発環境で実行できます。

(2) アプリケーション開発環境の特長

Cosminexus で構築するアプリケーションの開発環境の特長について説明します。

アプリケーション開発の一連の流れを開発環境上で実現

IDE である Eclipse および Eclipse のプラグインを使用して、アプリケーション開発時のプログラミング、テスト、デバッグの一連の作業を実施できます。

また、Cosminexus の開発環境では、Eclipse のプラグインをまとめた統合開発環境である、MyEclipse を提供しています。これによって、アプリケーション開発、テスト環境上でのテスト、デバッグなどの操作を、すべて Eclipse 上で実行できます。

1.2.5 サービス統合環境の概要と特長

ここでは、Cosminexus のサービス統合環境の概要と特長について説明します。

Cosminexus のサービス統合環境では、Cosminexus サービスプラットフォームを中心に、メッセージの送受信によるサービスの統合を実現します。

ここでは、概要および特長について説明します。Cosminexus サービスプラットフォー

ムの特長については、実行・運用環境と開発環境で分けて説明します。なお、サービス統合環境の詳細については、「7.1 サービス統合環境の特長」を参照してください。

(1) Cosminexus サービスプラットフォームの概要

Cosminexus では、ビジネス環境の変化に柔軟に対応するためのシステムとして、SOA の概念に対応したシステムを構築できます。SOA では、サービスという概念を基本として、システムを構築します。SOA で扱うサービスは、業務処理の一つの単位と考えられます。例えば、アプリケーション実行環境で動作する J2EE アプリケーションの一つ一つをそれぞれサービスとみなしたり、データベースに格納されているデータそのものをサービスとみなしたりできます。

Cosminexus では、これらの複数のサービスを新しいサービスとして活用するためのサービス統合環境を提供します。サービス統合環境では、新規に開発したサービスのほか、既存のアプリケーションなどをサービスとして扱うこともできます。また、複数のアプリケーションに順序性を持たせて組み合わせ、一つのサービスとして定義することもできます。

(2) Cosminexus サービスプラットフォームの特長（実行環境・運用環境）

ここでは、Cosminexus サービスプラットフォームの実行環境および運用環境の特長について説明します。

SOA の概念に基づくサービスインテグレーションの実現

既存のアプリケーション、新しく開発したアプリケーション、ほかのシステムで提供される多様なサービスなどを、SOA の概念で統合して利用するための基盤となる実行環境（Cosminexus サービスプラットフォーム）を構築・運用できます。また、サービスインテグレーションを実現するためのコンポーネントを定義・開発する環境や、実行環境を運用するための環境も構築できます。

レガシーシステムや ERP パッケージの統合

Cosminexus サービスプラットフォームで統合するサービスとして、OpenTP1 や Object Wrapper システムなどのレガシーシステム、および ERP パッケージで構築したシステムを利用できます。これによって、信頼性の高い既存資産を有効に活用できます。

(3) Cosminexus サービスプラットフォームの特長（開発環境）

ここでは、Cosminexus サービスプラットフォームの開発環境の特長について説明します。

サービスコンポーネント開発の効率化

Cosminexus サービスプラットフォームの開発環境では、次に示す手段によって、サービスコンポーネントの開発を効率化します。

- サービスコンポーネントの開発環境を一括して構築できます。

1. Cosminexus の概要

- ビジネスモデリングツールでエクスポートした BPEL 定義をインポートして使用できます。
- Eclipse 上でビジネスプロセスを定義できます。

1.3 Cosminexus で構築できる環境

この節では、Cosminexus で構築できる環境の組み合わせで構築できるシステムの概要について説明します。

また、アプリケーション実行環境とアプリケーション開発環境の関係、サービス統合環境の開発環境、運用環境および実行環境の関係についても説明します

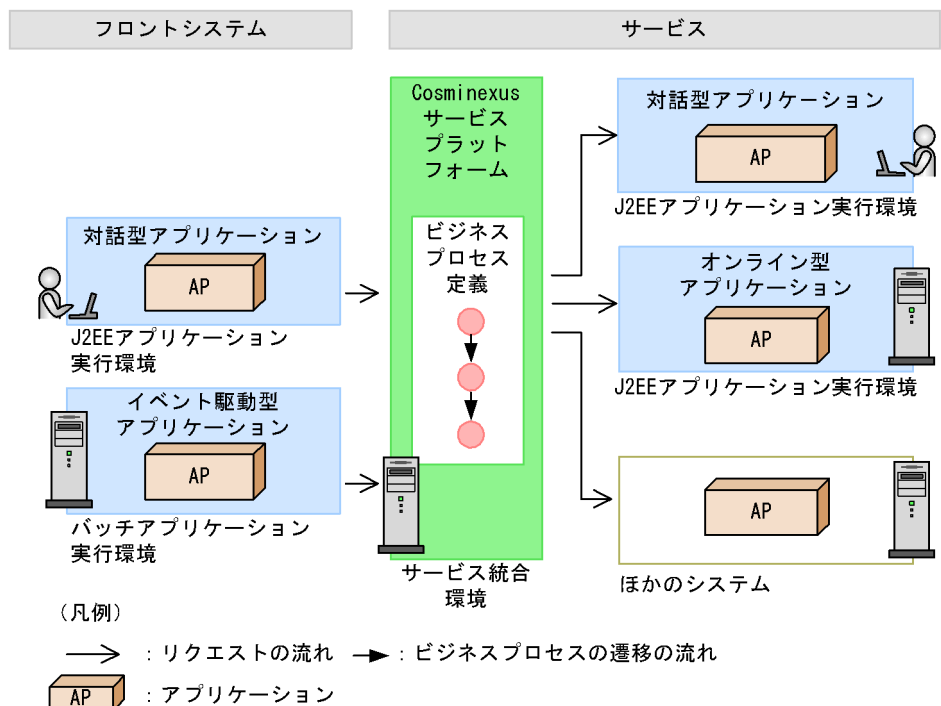
1.3.1 Cosminexus の実行環境を組み合わせたシステムの例

Cosminexus の実行環境を組み合わせたシステムを例に、それぞれの環境の位置づけを説明します。

Cosminexus の実行環境には、アプリケーション実行環境および Cosminexus サービスプラットフォームがあります。それぞれ、実行できるソフトウェアコンポーネントが異なります。

Cosminexus の実行環境を組み合わせたシステムの例を次の図に示します。この例は、Cosminexus サービスプラットフォームを中心として、サービスの統合を実現するシステムの例です。統合されるサービスの実行環境や、サービスを呼び出すフロントシステム側の実行環境に、Cosminexus のそれぞれの実行環境を配置しています。

図 1-5 Cosminexus の実行環境を組み合わせたシステムの例



1. Cosminexus の概要

ここでは、図で示した各環境の位置づけについて、フロントシステムとサービスに分けて説明します。フロントシステムは、Web ブラウザなどのフロントエンドから送信されるリクエストを受け付けて、そのリクエストを処理するシステムです。サービスは、フロントシステムから呼び出されて、リクエストの内容に応じた業務を実行します。

(1) フロントシステムを構成する実行環境

フロントシステムを構成する実行環境で稼働するソフトウェアコンポーネントについて説明します。

フロントシステムでは、次のソフトウェアコンポーネントが稼働します。

- 対話型アプリケーション
- イベント駆動型アプリケーション

それぞれのソフトウェアコンポーネントについて説明します。

(a) 対話型アプリケーション (J2EE アプリケーション実行環境)

対話型アプリケーションは、一人の作業者が、画面などのインタフェースを使用して、データを入力または出力しながら業務を遂行する形式のアプリケーションです。

対話型アプリケーションは、J2EE アプリケーション実行環境で稼働します。

(b) イベント駆動型アプリケーション (バッチアプリケーション実行環境)

イベント駆動型アプリケーションは、イベントが実行されたタイミングで、実行されるアプリケーションです。例えば、決まった時刻になったときやファイルが到着したときなどに実行されます。

イベント駆動型アプリケーションは、バッチアプリケーション実行環境と JP1/AJS2 などの関連プログラムと連携させた環境で稼働します。

(2) サービスを実現する実行環境

サービスを実現する実行環境で稼働するソフトウェアコンポーネントについて説明します。

次のソフトウェアコンポーネントが稼働します。

Cosminexus サービスプラットフォーム

対話型アプリケーション

オンライン型アプリケーション

このほか、ほかのシステム上で稼働しているソフトウェアコンポーネントにも、サービスとして利用できるものがあります。

(a) Cosminexus サービスプラットフォーム (サービス統合環境)

Cosminexus サービスプラットフォームは、複数のサービスを組み合わせて統合する環

境を表すソフトウェアコンポーネントです。BPEL を使用して定義されたビジネスプロセス定義に従って、サービス連携やメッセージ送受信を実現します。

Cosminexus サービスプラットフォームは、サービス統合環境で稼働します。

(b) 対話型アプリケーション (J2EE アプリケーション実行環境)

対話型アプリケーションは、一人の作業者が、画面などのインタフェースを使用して、データを入力または出力しながら業務を遂行する形式のアプリケーションです。また、サービスとして使用される対話型アプリケーションは、Cosminexus サービスプラットフォームとの連携するためのインタフェースも備えています。

対話型アプリケーションは、J2EE アプリケーション実行環境で稼働します。

(c) オンライン型アプリケーション (J2EE アプリケーション実行環境)

オンライン型アプリケーションは、呼び出し元からの問い合わせに対する応答処理によって、サービスを実現するアプリケーションです。例えば、Web サービスなどが該当します。呼び出し元に対しては、必要に応じて応答を返します。また、サービスとして使用されるオンライン型アプリケーションは、Cosminexus サービスプラットフォームと連携するためのインタフェースも備えています。

オンライン型アプリケーションは、J2EE アプリケーション実行環境で稼働します。

(d) ほかのシステム

ほかのシステムとして動作している、レガシーシステム、データベースまたは ERP パッケージなどです。Cosminexus サービスプラットフォーム上のビジネスプロセス定義と連携するためのインタフェースも備えています。Cosminexus サービスプラットフォームの機能で連携するため、Cosminexus の実行環境である必要はありません。

1.3.2 アプリケーションの実行環境と開発環境の関係

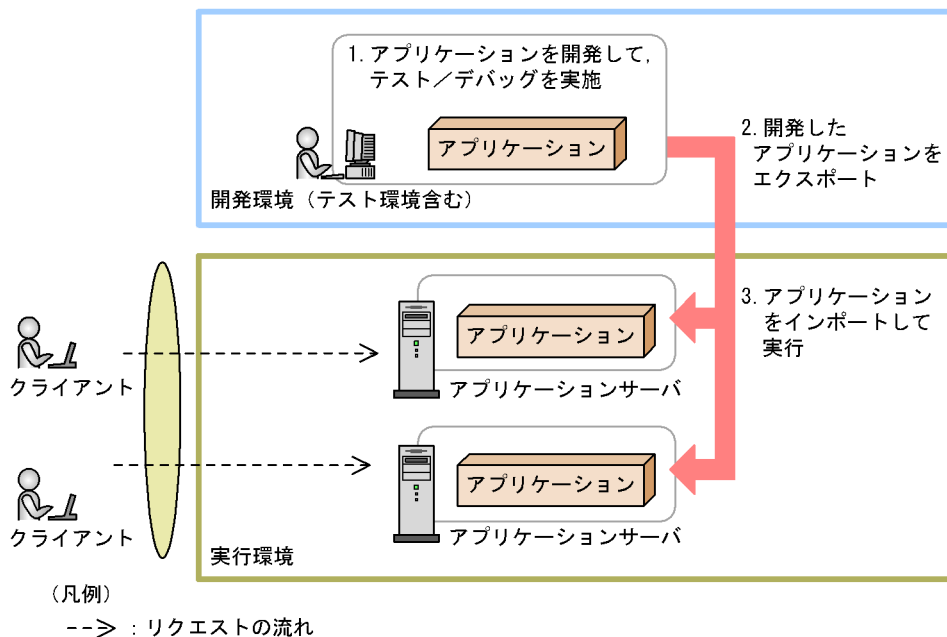
ここでは、アプリケーションの実行環境と開発環境の関係について説明します。

アプリケーションの開発環境では、対話型アプリケーションやオンライン型アプリケーションなどのアプリケーションを開発できます。

Cosminexus のアプリケーションの実行環境と開発環境の関係を、次の図に示します。

1. Cosminexus の概要

図 1-6 Cosminexus のアプリケーションの実行環境と開発環境の関係



図の流れについて説明します。

1. 開発環境で、アプリケーションを開発して、テスト/デバッグを実施します。
2. 1. で開発したアプリケーションをエクスポートします。
3. 2. でエクスポートしたアプリケーションを実行環境にインポートします。

なお、この図では、可用性を高めるために、一つのアプリケーションを複数の実行環境にインポートしています。インポート後、必要な設定およびカスタマイズを実行してから、実運用を開始します。

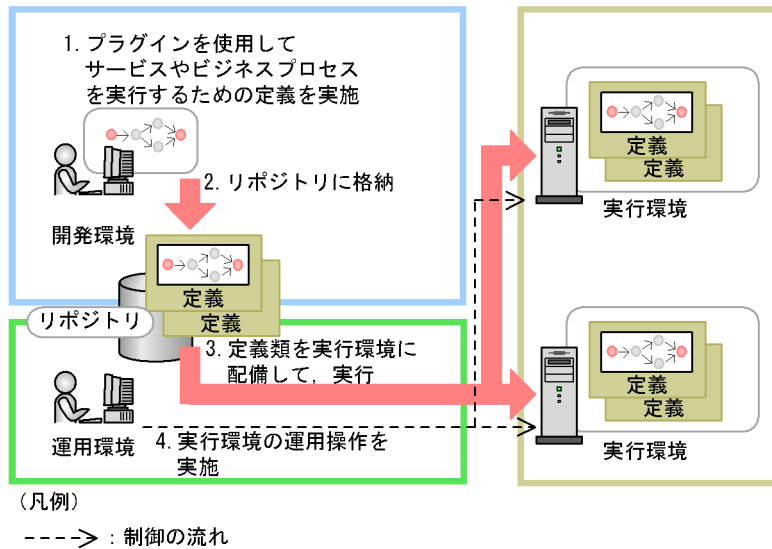
1.3.3 サービス統合環境の開発環境，運用環境および実行環境の関係

ここでは、サービス統合環境の開発環境，運用環境および実行環境の関係について説明します。

サービス統合環境の開発環境では、Cosminexus サービスプラットフォームによって統合するサービスやビジネスプロセスを定義できます。

Cosminexus サービスプラットフォームの開発環境，実行環境および運用環境の関係を、次の図に示します。

図 1-7 Cosminexus サービスプラットフォームの開発環境，実行環境および運用環境の関係



図の流れについて説明します。

1. 開発環境で，Eclipse に組み込んだプラグインを使用して，サービスと接続したり，サービスを組み合わせるため，各種定義を作成します。
2. 定義をアーカイブファイルにしてリポジトリに格納します。
3. 運用環境で，リポジトリの内容を実行環境に配備して，クライアントからサービスを利用できるように，各種定義を実行します。
4. 運用開始後，実行環境の運用操作を，運用環境から実施します。

参考

この説明では，流れを簡略化して説明しています。実際の環境では，運用環境から開発環境に取り込む必要がある情報などもあります。詳細は，マニュアル「Cosminexus サービスプラットフォーム 概説」を参照してください。

2

Cosminexus の製品構成

この章では、Cosminexus の製品構成について説明します。Cosminexus には、実行環境を構築する製品、開発環境を構築する製品、サービス統合環境を構築する製品、およびそれらの環境で必要に応じて使用するオプション製品があります。また、これらの製品は、構成ソフトウェアと呼ばれる複数のソフトウェアで構成されています。

2.1 製品の分類と特長

2.2 製品の概要

2.3 構成ソフトウェア

2.4 動作環境

2.1 製品の分類と特長

Cosminexus の製品は、用途ごとに、4 種類に分類できます。製品の分類を次の表に示します。

表 2-1 Cosminexus の製品の分類

分類	製品
実行環境を構築する製品	<ul style="list-style-type: none"> • Application Server Standard • Application Server Enterprise
開発環境を構築する製品	<ul style="list-style-type: none"> • Developer Standard • Developer Professional
サービス統合環境を構築する製品	実行環境 <ul style="list-style-type: none"> • Service Platform 開発環境 <ul style="list-style-type: none"> • Service Architect
オプション製品	<ul style="list-style-type: none"> • uCosminexus Client • uCosminexus Web Redirector • uCosminexus Operator

注 Service Platform は、サービス統合環境の運用環境でも使用します。

この節では、分類ごとの特長について説明します。

2.1.1 実行環境を構築する製品

J2EE に準拠した J2EE アプリケーションを、サーバサイドで実行するための実行環境を構築するための製品です。Web サービスに対応した SOAP アプリケーションや、メッセージによる非同期通信に対応したアプリケーションも実行できます。クライアントには、Web ブラウザ、または EJB クライアントアプリケーションを使用できます。

実行環境を構築する製品を次に示します。

Application Server Standard

Application Server Enterprise

Application Server Enterprise は、Application Server Standard に比べて、バックエンドシステムとしての機能と、セキュリティ管理のための機能に優れています。また、Application Server Enterprise では、バッチアプリケーションを実行する環境も構築できます。

なお、実行環境では、システム構成に応じてオプション製品も利用できます。

2.1.2 開発環境を構築する製品

Cosminexus の実行環境で実行する J2EE アプリケーションの開発環境を構築するための製品です。IDE などを使用して開発した J2EE アプリケーションに対して、アセンブル、デバッグ、デプロイ、テストなどができます。

また、J2EE アプリケーションの開発に Eclipse を使用する場合、MyEclipse を使用できます。MyEclipse は、Eclipse 上でのアプリケーション開発で使用するプラグインをまとめた統合開発環境です。MyEclipse を使用すると、UML の設計からコーディング、テスト、デバッグまで、一連の操作をまとめて実行できます。

開発環境を構築する製品を次に示します。

Developer Standard

Developer Professional

Developer Standard は、一つの J2EE サーバ内で動作するアプリケーションを開発するための製品です。Developer Professional は、別プロセスや別マシン上のアプリケーションを呼び出すアプリケーションを開発したり、cFramework を使用してアプリケーションを開発したりするための製品です。

なお、Developer Standard および Developer Professional は、Windows 環境だけで使用できます。

2.1.3 サービス統合環境を構築する製品

実行環境と開発環境の機能をベースにして、業務システムをビジネス環境の変化に応じて柔軟に再編成するための基盤（Cosminexus サービスプラットフォーム）を実現する製品です。

Cosminexus サービスプラットフォームでは、個々の業務に対応した J2EE アプリケーションをサービスとして扱います。これらのサービスに対して統一されたインタフェースでアクセスしたり、既存のサービスや新規に開発したサービスを組み合わせて新たなサービスを作成したりできます。

サービス統合環境の実行環境または運用環境を構築するための製品を次に示します。

Service Platform

サービス統合環境の実行環境または運用環境を構築するための製品です。

Service Architect

サービスを利用するための定義などを実行するための製品です。Windows 環境だけで使用できます。

なお、実行環境では、必要に応じてオプション製品も利用できます。

2.1.4 オプション製品

Cosminexus で実行環境を構築する場合に、システム構成に応じて使用できる製品です。

業務システム内に、用途を限定したマシンを配置する場合に使用できます。

オプション製品を次に示します。

uCosminexus Client

実行環境上の Enterprise Bean をクライアントマシンのアプリケーションから直接呼び出す構成の場合に使用できます。また、SOAP アプリケーションの実行環境上の SOAP サービスを、クライアントマシンのアプリケーションから RPC を使用して呼び出す構成の場合にも使用できます。

Windows 環境だけで使用できます。

uCosminexus Web Redirector

Web サーバとアプリケーションサーバを別のマシンに分けて配置する場合に使用できます。

uCosminexus Operator

実行環境と運用環境を別のマシンに分けて配置する場合に使用できます。UNIX 実行環境に対して Eclipse のプラグインを使用した運用操作をしたい場合は、必ず使用します。Windows 環境だけで使用できます。

2.2 製品の概要

この節では、次の製品の概要について説明します。

実行環境を構築する製品

- Application Server Standard
- Application Server Enterprise

開発環境を構築する製品

- Developer Standard
- Developer Professional

サービス統合環境を構築する製品

- Service Platform
- Service Architect

オプション製品

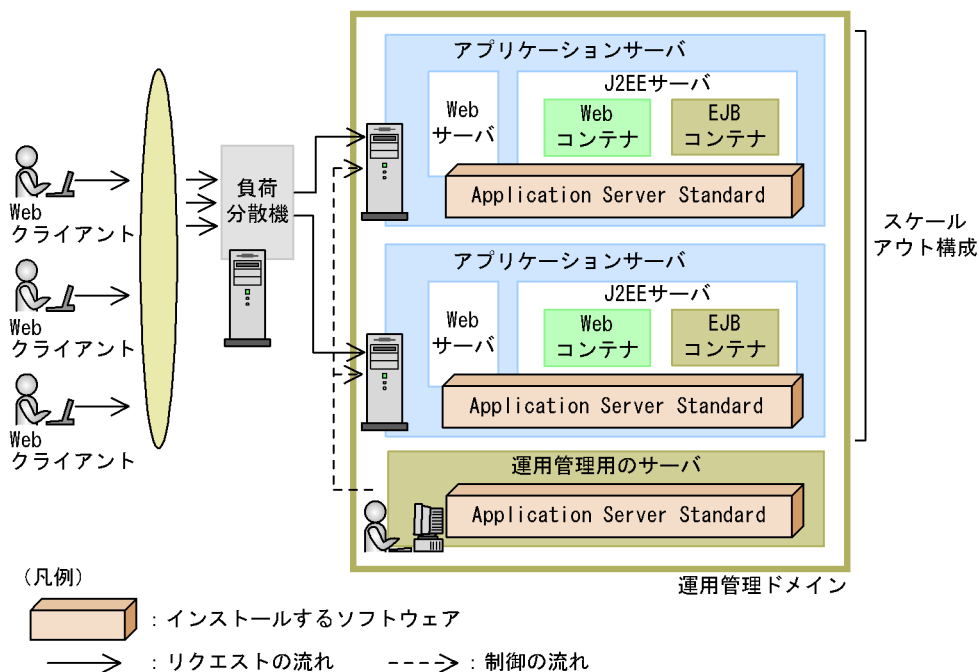
- uCosminexus Client
- uCosminexus Web Redirector
- uCosminexus Operator

2.2.1 Application Server Standard

Application Server Standard は、実行環境を構築するための製品です。スケールアウト構成のシステムを構築・運用するのに適しています。スケールアウト構成とは、同じ処理をするサーバマシンを増やしてリクエスト処理を実行することで、システムの処理性能を高め、かつ可用性や信頼性を確保できる構成です。

Application Server Standard で構築できるスケールアウトシステムの構成例を次の図に示します。

図 2-1 Application Server Standard で構築できるスケールアウトシステムの構成例



スケールアウト構成の場合、Web クライアントからのリクエストを負荷分散機が受け付け、アプリケーションサーバに振り分けます。

Application Server Standard では、スケールアウト構成にしたシステムの一括構築や一括運用を実現できます。また、アプリケーションサーバを運用管理ドメインという範囲で一括管理して、ドメイン内の個別のサーバマシンに対するリモート操作での運用管理も実現できます。

Application Server Standard は、アプリケーションサーバにするマシンにインストールします。業務システムを一括構築・運用する場合に、運用管理用のサーバを別のマシンとして構築するときには、運用管理用のサーバにするマシンにも Application Server Standard をインストールします。

2.2.2 Application Server Enterprise

Application Server Enterprise は、実行環境を構築するための製品です。オンライン処理を実行するアプリケーションや、バッチ処理を実行するアプリケーションの実行環境を構築できます。

Application Server Standard に比べて、次の機能が強化されています。

Enterprise Bean のスケジューリング（流量制御 / 優先制御 / 閉塞制御）と負荷分散の実現

Web サービスに対するセキュリティ管理

バッチアプリケーション実行基盤としての機能

ここでは、J2EE アプリケーションの実行環境を構築する場合と、バッチアプリケーションの実行環境を構築する場合に分けて説明します。

(1) Application Server Enterprise を使用して構築できる J2EE アプリケーションの実行環境

J2EE アプリケーションの実行環境を構築する場合、Application Server Enterprise は、フロントエンド/バックエンドで構成される業務システムを構築・運用する場合に適しています。フロントエンド/バックエンドシステムとは、業務システム内のサーバを、用途ごとに、次のようにフロントエンドとバックエンドに分けて運用するシステム構成です。

フロントエンド

Web ブラウザから送信されるリクエストを受け付ける部分です。受け付けたリクエストをバックエンドシステムの業務サービスに振り分けたり、バックエンドシステムからのレスポンスを Web ブラウザに送信したりします。

アプリケーションとして、サーブレット、JSP、Enterprise Bean などが動作します。

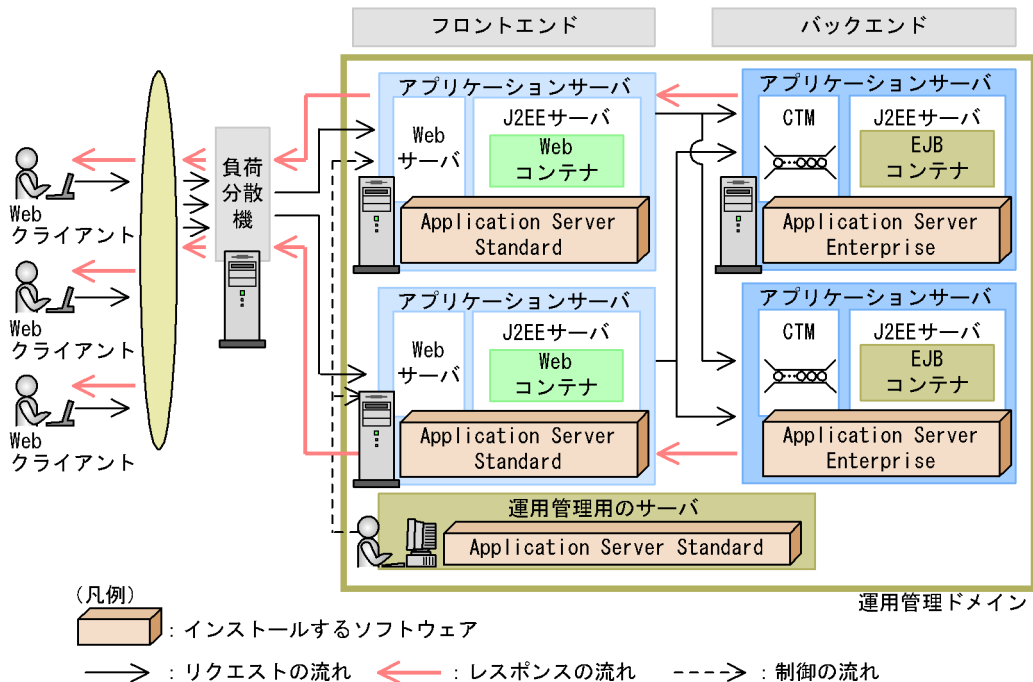
バックエンド

フロントエンドの背後で動作する、複数の業務に共通な業務サービス処理を実行するシステムです。Enterprise Bean などの業務処理プログラムが動作します。データベースアクセスなども実行します。

バックエンドシステムを Application Server Enterprise を使用して構築することによって、Enterprise Bean に対するスケジューリングや負荷分散などの機能を生かした、安定稼働性および可用性の高いシステムを構築できます。

Application Server Enterprise で構築できるフロントエンドシステム/バックエンドシステムの構成例を次の図に示します。

図 2-2 Application Server Enterprise で構築できるフロントエンド/バックエンドシステムの構成例



フロントエンド/バックエンドシステム構成の場合，Web クライアントからのリクエストを負荷分散機が受け付け，フロントエンドのアプリケーションサーバに振り分けます。フロントエンドのアプリケーションサーバは，バックエンドのアプリケーションサーバにリクエストを振り分けます。バックエンドのアプリケーションサーバでは，EJB スケジューラとしての機能を持つ CTM によってリクエストの負荷分散が実行され，適切なバックエンドのアプリケーションサーバ上で業務処理プログラムが実行されます。

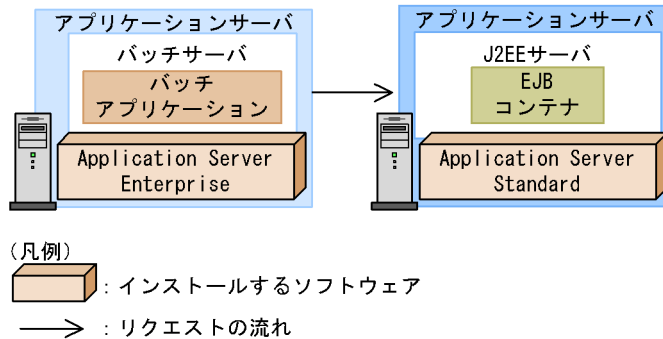
Application Server Enterprise で構築したシステムも，運用管理ドメインの範囲での一括管理ができます。

Application Server Enterprise は，バックエンドに配置するアプリケーションサーバのマシンにインストールします。また，フロントエンドのアプリケーションサーバ，または運用管理用のサーバにするマシンには，Application Server Standard または Application Server Enterprise をインストールします。

(2) Application Server Enterprise を使用して構築できるバッチアプリケーションの実行環境

Application Server Enterprise で構築できるバッチアプリケーションの実行環境の構成例を次の図に示します。

図 2-3 Application Server Enterprise で構築できるバッチアプリケーションを実行するシステムの構成例



バッチアプリケーションは、Application Server Enterprise をインストールして構築した環境で実行します。バッチアプリケーションから Enterprise Bean を呼び出す場合、Enterprise Bean は Application Server Standard または Application Server Enterprise をインストールして構築した環境の EJB コンテナで実行します。なお、バッチアプリケーションの実行環境では、運用管理機能はアプリケーションサーバと同じサーバで実行します。

2.2.3 Developer Standard

Developer Standard は、開発環境を構築するための製品です。J2EE アプリケーションの作成・テスト・デバッグといったアプリケーション開発の一連の流れを、スタンドアロン PC 内で実行できます。

ポイント

Developer Standard では、一つのマシン内の一つの J2EE サーバ内で動作する J2EE アプリケーションを開発できます。また、構築できるテスト用実行環境も、スタンドアロン PC での運用を前提としています。このため、開発した J2EE アプリケーションを複数のマシンで構成される実行環境上で実行することを想定していたり、J2EE アプリケーション内でプロセス間呼び出しを実行する J2EE アプリケーションを開発したりする場合は、Developer Professional を使用してください。

Developer Standard で構築した開発環境でのアプリケーション開発の特長は次のとおりです。

アプリケーションの開発に、Eclipse および Eclipse のプラグイン (MyEclipse など) を使用できます。

なお、テスト用実行環境は、実行環境をスケールアウトシステムで構築する場合と同様に、一括で構築できます。

2.2.4 Developer Professional

Developer Professional は、開発環境を構築するための製品です。Developer Standard の機能に加えて、次の機能を使用できます。

Cosminexus cFramework が提供するコンポーネント・フレームワークに基づいたアプリケーションの開発

プロセス間でのリモート呼び出しをする J2EE アプリケーションの開発

リモート呼び出しを想定したテスト用実行環境の構築

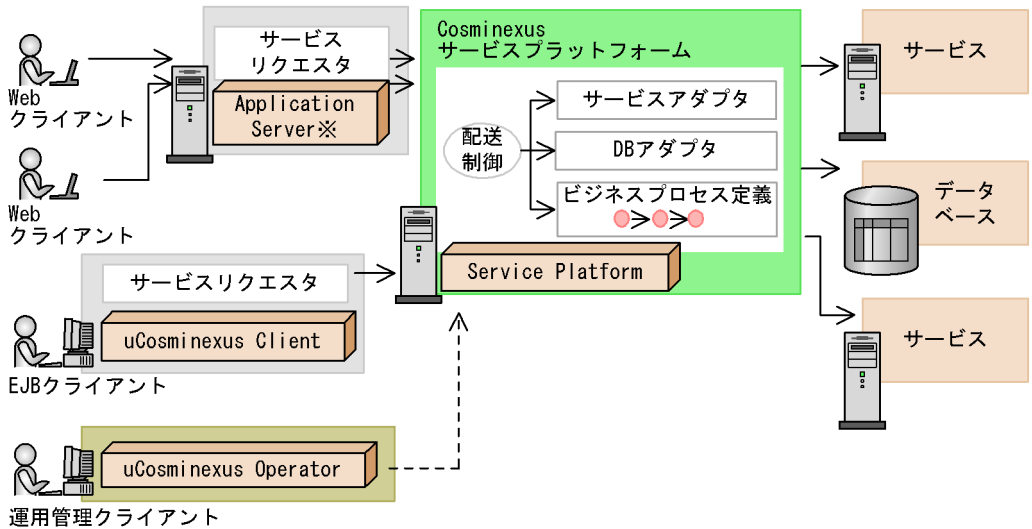
Developer Professional のテスト用実行環境では、Application Server Enterprise で構築した実行環境と同じ機能を使用して J2EE アプリケーションをテストできます。

2.2.5 Service Platform




Service Platform は、Cosminexus によるサービス統合を実現するための実行環境である、Cosminexus サービスプラットフォームを構築するための製品です。実行環境製品である Application Server Enterprise の機能に、サービス統合に必要な機能が追加されています。

Service Platform を利用した Cosminexus サービスプラットフォームの構築例を次の図に示します。

図 2-4 Cosminexus サービスプラットフォームの構成例



(凡例)

 : インストールするソフトウェア
 : リクエストの流れ  : 制御の流れ

注※ Webクライアントからのリクエストを受け付けてCosminexus サービスプラットフォームにリクエストを送信するサーバです。Cosminexus以外のJava実行環境も使用できます。

クライアントからのサービスに対するリクエストは、Cosminexus サービスプラットフォームが受け付けます。Cosminexus サービスプラットフォームでは、リクエストを各サービスが存在するサーバに送信します。サービスへの接続には、接続するサービスの種類ごとのアダプタを使用します。なお、Cosminexus サービスプラットフォームにリクエストを送信するプログラムをサービスリクエスタといいます。サービスリクエスタは、アプリケーションサーバやEJBクライアント上で実行します。

Service Platform は、Cosminexus サービスプラットフォームの実行環境にするマシンにインストールします。また、サービスリクエスタを Web アプリケーションとして開発する場合は、Web ブラウザからのリクエストを受け付けて Cosminexus サービスプラットフォームに送信するサーバとして、Application Server Standard または Application Server Enterprise をインストールしたアプリケーションサーバを利用できます。

参考

オプション製品について

EJB クライアントからリクエストを送信する場合は、EJB クライアントにするマシンに uCosminexus Client をインストールします。Cosminexus サービスプラットフォームの運用操作を別のマシンから実行したい場合は、運用管理用のクライアントに uCosminexus Operator をインストールします。これらの製品については、「2.2.7 uCosminexus Client」および「2.2.9 uCosminexus Operator」を参照してください。

2.2.6 Service Architect

Service Architect は、Cosminexus サービスプラットフォームで扱うサービスやビジネスを実行するために必要なコンポーネントを定義する環境を構築する製品です。開発環境製品である Developer Professional の機能に、コンポーネントを定義するための機能が追加されています。

Service Architect では、Eclipse のプラグインを使用して、定義を視覚的に作成できません。上流工程でビジネスプロセスを設計する際に、ビジネスモデリングツールを使用して設計している場合は、ツールで作成したビジネスプロセス設計図（BPEL 定義）を Service Architect の環境に取り込んで、使用できます。

なお、作成した定義を確認するためのテスト用実行環境として、Service Platform で構築した場合と同じ機能を持つ実行環境を構築できます。

2.2.7 uCosminexus Client

uCosminexus Client は、次に示すクライアントの実行環境を構築するためのオプション製品です。

EJB クライアントアプリケーションの実行環境

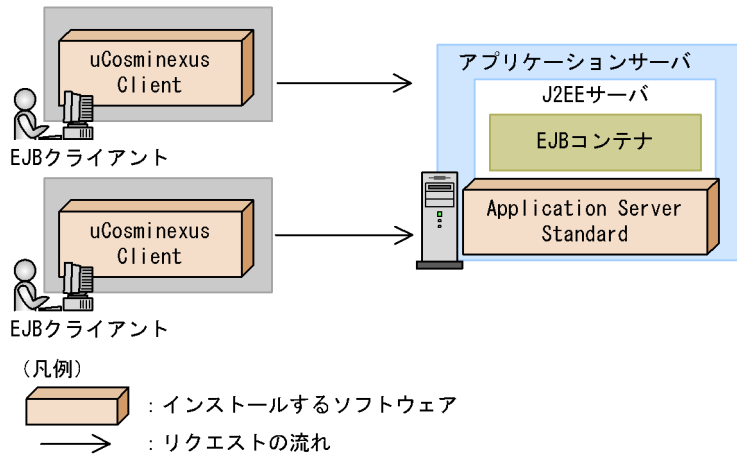
アプリケーションサーバ上で動作している J2EE アプリケーション内の Enterprise Bean を、Web サーバ経由ではなく、クライアントマシン上のプログラムから RMI-IIOP 通信で直接呼び出すシステムを構築する場合に使用します。

SOAP アプリケーションのクライアントになる Java アプリケーションの実行環境

アプリケーションサーバ上で動作している SOAP サービスを、クライアントマシン上の Java アプリケーションから、RPC を使用して呼び出す場合に使用します。

uCosminexus Client を利用したシステム構成の例を次の図に示します。この図は、EJB クライアントアプリケーションの実行環境の場合の例です。

図 2-5 uCosminexus Client を利用したシステム構成の例



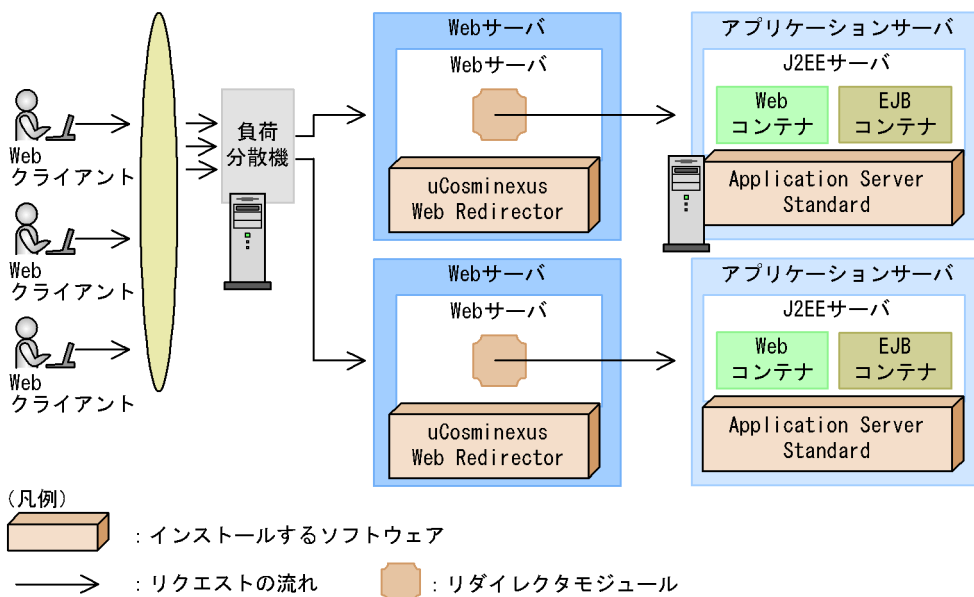
uCosminexus Client をインストールした EJB クライアントマシンからは、RMI-IIOP 通信によって、直接 J2EE サーバ上の Enterprise Bean にアクセスします。アプリケーションサーバに、構成ソフトウェアとして Cosminexus Component Transaction Monitor を含む製品がインストールされている場合は、EJB スケジューラである CTM 経由でもアクセスできます。製品と構成ソフトウェアの対応については、「2.3.1 製品と構成ソフトウェアの対応」を参照してください。

2.2.8 uCosminexus Web Redirector

uCosminexus Web Redirector は、実行環境を構築する場合に、Web サーバとアプリケーションサーバを異なるマシンに配置するとき使用するオプション製品です。Web サーバを配置したマシンに uCosminexus Web Redirector をインストールして、Web サーバにアプリケーションサーバ上の Web コンテナと連携するためのモジュール（リダイレクタモジュール）を組み込んで動作させます。

uCosminexus Web Redirector を利用したシステム構成の例を次の図に示します。

図 2-6 uCosminexus Web Redirector を利用したシステム構成の例



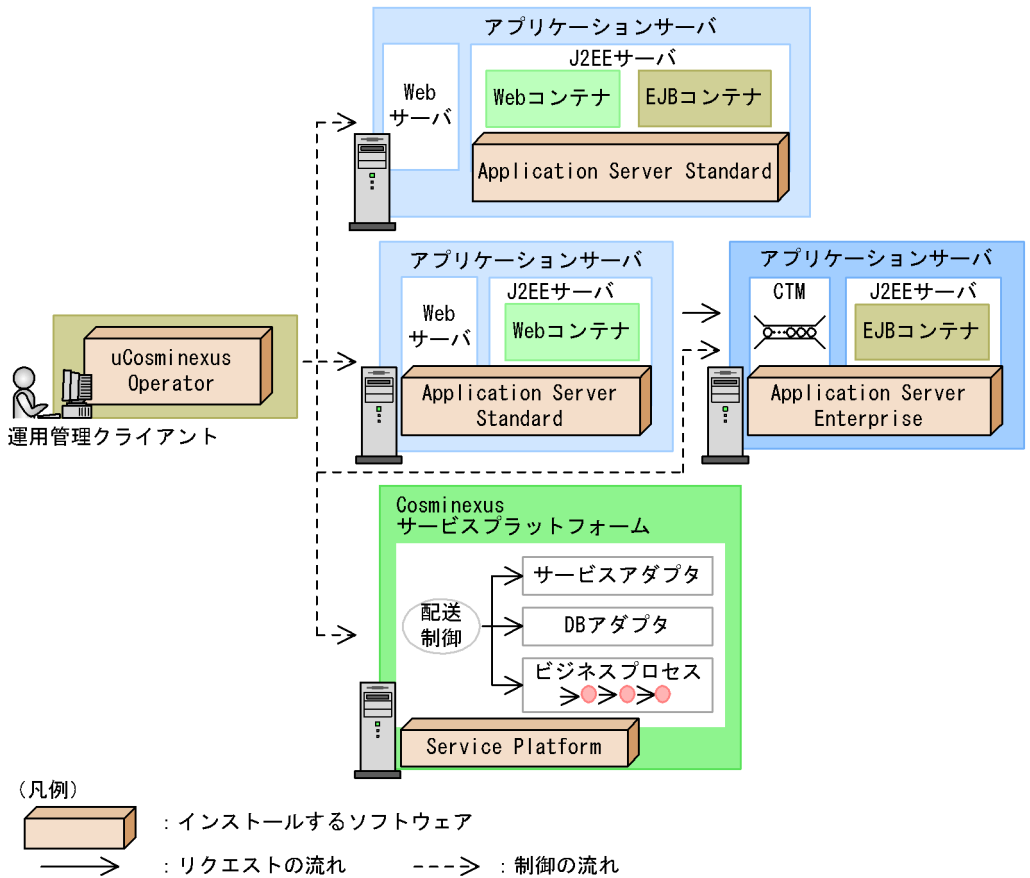
uCosminexus Web Redirector は、Web サーバと同じマシンにインストールします。Web サーバが受け付けたリクエストは、リダイレクタモジュールを経由して、アプリケーションサーバの Web コンテナに送信されます。

2.2.9 uCosminexus Operator

uCosminexus Operator は、Application Server Standard、Application Server Enterprise または Service Platform をインストールした実行環境のマシンとは別の Windows マシンから、GUI (Eclipse のプラグイン) を使用して実行環境を操作するためのオプション製品です。実行環境が UNIX の場合に Eclipse プラグインの GUI を使用した運用操作を実行したいときには、この製品が必要です。また、Service Platform をインストールした実行環境に対しては、セットアップなどを実行する CUI も使用できます。

uCosminexus Operator を利用したシステム構成の例を次の図に示します。

図 2-7 uCosminexus Operator を利用したシステム構成の例



uCosminexus Operator をインストールしたマシンから、アプリケーションサーバまたは Cosminexus サービスプラットフォームをインストールしたマシンの運用操作を実行します。

2.3 構成ソフトウェア

この節では、Cosminexus 製品を構成する構成ソフトウェアについて説明します。構成ソフトウェアは、単体で動作させるのではなく、ほかの構成ソフトウェアの機能と組み合わせて動作させることによって、アプリケーションサーバとしての機能を実現します。

構成ソフトウェアは、Cosminexus の機能に対応します。製品に含まれている構成ソフトウェアの種類によって、実現できる機能が異なります。

2.3.1 製品と構成ソフトウェアの対応

ここでは、製品と、その製品に含まれる構成ソフトウェアの対応を示します。

製品に対応する構成ソフトウェアを、次の表に示します。

表 2-2 製品と構成ソフトウェアの対応

構成ソフトウェア名	実行環境を構築する製品		開発環境を構築する製品		サービス統合環境を構築する製品	
	Application Server Standard	Application Server Enterprise	Developer Standard	Developer Professional	Service Platform	Service Architect
Cosminexus Application Development Plug-in	-	-			-	
Cosminexus cFramework			-			
Cosminexus Component Container						
Cosminexus Component Transaction Monitor	-		-			
Cosminexus DABroker Library						
Cosminexus Developer's Kit for Java						
Cosminexus Performance Tracer						
Cosminexus Reliable Messaging			-			

構成ソフトウェア名	実行環境を構築する製品		開発環境を構築する製品		サービス統合環境を構築する製品	
	Application Server Standard	Application Server Enterprise	Developer Standard	Developer Professional	Service Platform	Service Architect
Cosminexus Service Coordinator	-	-	-	-		
Cosminexus Service Development Plug-in	-	-	-	-	-	
Cosminexus Studio cFramework	-	-		-	-	-
Cosminexus TPBroker			-			
Cosminexus Web Services - Security	-		-			
Cosminexus XML Processor						
HiRDB Embedded Server Version 8	-	-			-	
Hitachi Web Server						
My Eclipse for Cosminexus	-	-			-	

(凡例)

: 含まれます。 - : 含まれません。

オプション製品に対応する構成ソフトウェアを、次の表に示します。

表 2-3 製品と構成ソフトウェアの対応 (オプション製品)

構成ソフトウェア名	オプション製品		
	uCosminexus Client	uCosminexus Web Redirector	uCosminexus Operator
Cosminexus Component Container - Client		-	-

2. Cosminexus の製品構成

構成ソフトウェア名	オプション製品		
	uCosminexus Client	uCosminexus Web Redirector	uCosminexus Operator
Cosminexus Component Container - Redirector	-		-
Cosminexus Developer's Kit for Java		-	
Cosminexus Operator Plug-in	-	-	
Cosminexus Performance Tracer			-
Cosminexus TPBroker		-	-
Cosminexus XML Processor		-	

(凡例)

:含まれます。 - :含まれません。

2.3.2 構成ソフトウェアの機能概要

それぞれの構成ソフトウェアの機能概要について説明します。

(1) Cosminexus Application Development Plug-in

開発環境で使用する次の機能を提供する構成ソフトウェアです。

Cosminexus コネクタ (Eclipse プラグイン)

開発環境の MyEclipse から、テスト環境上の J2EE サーバを操作するための機能です。Eclipse プラグインとして提供します。

開発環境のインスタントセットアップ機能

ウィザードプログラム (GUI) によってアプリケーションの開発環境およびテスト環境をセットアップできる機能です。

(2) Cosminexus cFramework

コンポーネント・フレームワークに基づいて開発されたアプリケーションコンポーネントを実行する構成ソフトウェアです。

(3) Cosminexus Component Container

アプリケーションの実行基盤の中核として、次のような機能を提供する構成ソフトウェアです。

J2EE アプリケーションの実行環境 (J2EE サーバ) としての機能

バッチアプリケーションの実行環境（バッチサーバ）としての機能

SOAP アプリケーションの実行・開発環境としての機能

アプリケーションサーバを運用管理する機能

Web アプリケーション開発のためのコンポーネント提供機能

それぞれの機能の概要を示します。

(a) J2EE アプリケーションの実行環境（J2EE サーバ）としての機能

サーバサイドの業務処理プログラム（ビジネスロジック）をコンポーネントとして実行するためのフレームワークである，J2EE サーバを実現するための機能です。Web コンテナ，EJB コンテナなどの機能を含みます。Sun Microsystems, Inc. が提唱する Java™ 2 Platform, Enterprise Edition (J2EE) 1.4 に含まれる仕様に準拠しています。

J2EE 1.4 に含まれる仕様を次に示します。

Enterprise JavaBeans™ (EJB) 2.1

Java™ Servlet 2.4

JavaServer Pages™ (JSP) 2.0

JDBC™ 3.0 API

JDBC 3.0 Standard Extension API

Java Naming and Directory Interface™ (JNDI) 1.2

Java Transaction API (JTA) 1.0.1B

JavaMail™ API 1.3

Java-To-IDL Mapping (RMI-IIOP 1.0)

J2EE Connector Architecture (JCA) 1.0 および 1.5

Java Message Service (JMS) 1.1

さらに，構成ソフトウェアに Cosminexus Component Transaction Monitor を含む製品の場合は，CTM 機能と連携して，Enterprise Bean に対する動的負荷分散，流量制御，優先制御およびサービス閉塞に対応するための基盤機能も提供します。

また，J2EE サーバ上で動作するアプリケーションまたはリソースを操作するための Eclipse 連携用のプラグインである，Server Plug-in も提供します。

(b) バッチアプリケーションの実行環境（バッチサーバ）としての機能

バッチアプリケーションをサーバで実行するための機能です。バッチ処理の処理内容を Java で実装した Java アプリケーションを実行できます。サーバ上でアプリケーションを動作させることによって，Java VM の起動コストを抑えられます。コネクションプー

2. Cosminexus の製品構成

ルやステートメントプールを使用した効率の良いデータベースアクセスや、フルガーベージコレクションの制御なども実現できます。

なお、この機能は、Application Server Enterprise または Developer Professional を使用している場合にだけ使用できる機能です。

(c) SOAP アプリケーションの実行・開発環境としての機能

SOAP アプリケーションの実行環境および開発環境としての機能です。

SOAP アプリケーションの実行環境に必要な、SOAP エンジンと SOAP クライアントライブラリを提供します。

また、WSDL の作成支援機能や、SOAP 通信基盤対応のスタブおよびスケルトンの自動生成機能によって、SOAP アプリケーションの開発を支援します。

(d) アプリケーションサーバを運用管理する機能

アプリケーションサーバを運用管理するための機能です。次のような運用管理を実現できます。

アプリケーションサーバの一括構築・一括運用

J2EE サーバ内のアプリケーションやリソースの設定

複数の J2EE アプリケーションのユーザ管理機能をシームレスに連携した統合ユーザ管理

Cosminexus の各機能が出力するログの収集

また、JP1 などのほかの運用管理プログラム製品と連携して、Cosminexus の運用管理をするためのコマンドも提供しています。

(e) Web アプリケーション開発のためのコンポーネント提供機能

Web アプリケーションの開発時に利用できる、コンポーネントを提供する機能です。

Cosminexus Component Library は、Web アプリケーション開発の開発効率と品質を向上させるための機能を提供するコンポーネント群です。クロスサイトスクリプティングに対応するためのコンポーネントや、データベースのキャッシュを実現するコンポーネント、セッションデータ管理で使用するコンポーネントなどが含まれます。

(4) Cosminexus Component Container - Client

EJB クライアントアプリケーションの実行環境を構築するための構成ソフトウェアです。Cosminexus Component Container のサブセットです。

(5) Cosminexus Component Container - Redirector

Web コンテナを Web サーバと連携させるための構成ソフトウェアです。この構成ソフトウェアで提供されるリダイレクタモジュールを Web サーバに登録することで、Web サー

バあての HTTP リクエストのうち、特定のリクエストを指定した Web コンテナに処理させたり、複数の Web コンテナにリクエストを振り分けて処理させたりできます。

(6) Cosminexus Component Transaction Monitor

Enterprise Bean に対するクライアントからのリクエストをスケジューリングして、負荷分散や流量制御を実現する構成ソフトウェアです。アプリケーションごとにキューを管理して各サーバの負荷状況に応じて処理を分散させたり、一度に処理するリクエストの数を制御したり、業務処理プログラムを入れ替える時に特定の J2EE アプリケーションだけを閉塞させたりできます。これによって、システムが安定した状態で運転し続けることができるので、業務システムの可用性と信頼性が向上します。

(7) Cosminexus DABroker Library

データベース連携機能を提供する構成ソフトウェアです。JDBC インタフェースをサポートしたデータベースにアクセスするライブラリを提供します。これによって、複数の DBMS に一つのインタフェースでアクセスできる統一的なデータベースアクセスを実現できます。

(8) Cosminexus Developer's Kit for Java

Java 2 Platform Standard Edition 5.0 に準拠した構成ソフトウェアです。対応する Sun Microsystems 社製の JDK のバージョンは JDK 5.0 です。JDK 5.0 で使用できる機能、コマンドおよび API については、Sun Microsystems 社が提供している JDK 5.0 のドキュメントを参照してください。

(9) Cosminexus Operator Plug-in

アプリケーションの実行環境と運用操作環境を異なるマシンに分けて構築する場合に、uCosminexus Operator をインストールしたマシンから、Application Server Standard、Application Server Enterprise または Service Platform をインストールしたマシン上の J2EE サーバや HCSC サーバ (Cosminexus サービスプラットフォームの実行基盤) を操作するためのプラグインを提供する構成ソフトウェアです。

次の 2 種類の Eclipse プラグインを使用した GUI 操作を実現できます。

Server Plug-in

J2EE サーバ上の J2EE アプリケーションまたは J2EE リソースを操作するための Eclipse プラグインです。

HCSC-Manager Plug-in

Cosminexus サービスプラットフォームで利用する HCSC サーバを運用操作するための Eclipse プラグインです。

また、HCSC-Manager が提供する CUI を使用した運用操作も実行できます。

(10)Cosminexus Performance Tracer

処理性能のボトルネックを解析するためのトレース情報を出力する構成ソフトウェアです。Cosminexus で構築したシステムでは、リクエストが処理される時に、決められたポイントごとに性能解析用のトレース情報を出力します。この情報を収集して分析することで、システムのボトルネックが調査できます。また、障害が発生した場合には、障害の発生個所を特定することもできます。

(11)Cosminexus Reliable Messaging

業務コンポーネント間、社内システム間、および社内システムと社外システム間で、非同期に高信頼なメッセージ送受信を実現する高信頼メッセージング基盤としての機能を提供する構成ソフトウェアです。

(12)Cosminexus Service Coordinator

Cosminexus によってサービス統合を実現するための基盤機能 (Cosminexus サービスプラットフォーム) を提供する構成ソフトウェアです。

SOA の概念でシステム内のサービスを統合するための、メッセージング、ビジネスプロセス、データ変換、DB アダプタなどの機能を提供します。

(13)Cosminexus Service Development Plug-in

Cosminexus サービスプラットフォームの開発環境として、ビジネスプロセス、アダプタ類、データ変換などを定義する機能を提供する構成ソフトウェアです。

(14)Cosminexus Studio cFramework

コンポーネント・フレームワークに基づく、アプリケーションコンポーネントの開発環境を提供する構成ソフトウェアです。

(15)Cosminexus TPBroker

サーブレットまたは JSP と Enterprise Bean の間などの通信で使用される、J2EE での RMI-IIOP 通信基盤、および分散トランザクション基盤になる構成ソフトウェアです。また、J2EE 環境での CORBA クライアントアプリケーションの実行環境および開発環境を提供します。また、EJB クライアントアプリケーションの実行環境で使用する、RMI-IIOP 通信基盤、および CORBA クライアントアプリケーションの実行環境としての機能も提供します。

(16)Cosminexus Web Services - Security

Web サービスのセキュリティの標準規格 WS-Security に準拠した機能を提供する構成ソフトウェアです。SOAP メッセージに対して、XML 署名を付けたり、SOAP メッセージを暗号化したりします。Cosminexus Web Services - Security の利用によって、セキュアな状態で SOAP メッセージの送受信ができます。また、SOAP メッセージに付いてい

る署名を検証したり、暗号化された SOAP メッセージを復号化したりできます。

XML 署名の生成・検証、または XML 暗号によるデータの暗号化・復号化を行うアプリケーションの開発を支援する機能も提供しています。

(17) Cosminexus XML Processor

業界標準の JAXP1.3 をサポートした XML ドキュメントの読み取り、操作および生成を実行する構成ソフトウェアです。

(18) HiRDB Embedded Server Version 8

アプリケーション開発時、テストおよびデバッグに使用できるデータベースです。

(19) Hitachi Web Server

Apache2.0 をベースに Secure Sockets Layer (SSL) をサポートしたミッションクリティカル分野向けの Web サーバです。

(20) MyEclipse for Cosminexus

Eclipse プラグインをまとめた統合開発環境 MyEclipse の日本語版環境です。

Eclipse でアプリケーションを開発するときに使用するプラグインがまとめられています。アプリケーションの作成、J2EE サーバへのアプリケーションのデプロイ、J2EE サーバの起動・停止などができます。また、Struts や JSF などのフレームワークを使用したアプリケーションの開発や、UML によるアプリケーション設計もできます。

2.3.3 実行環境の構成ソフトウェアの位置づけ (Application Server)

ここでは、実行環境を構築する製品である Application Server (Application Server Standard または Application Server Enterprise) の各構成ソフトウェアの位置づけについて、実行するアプリケーションの種類ごとに説明します。

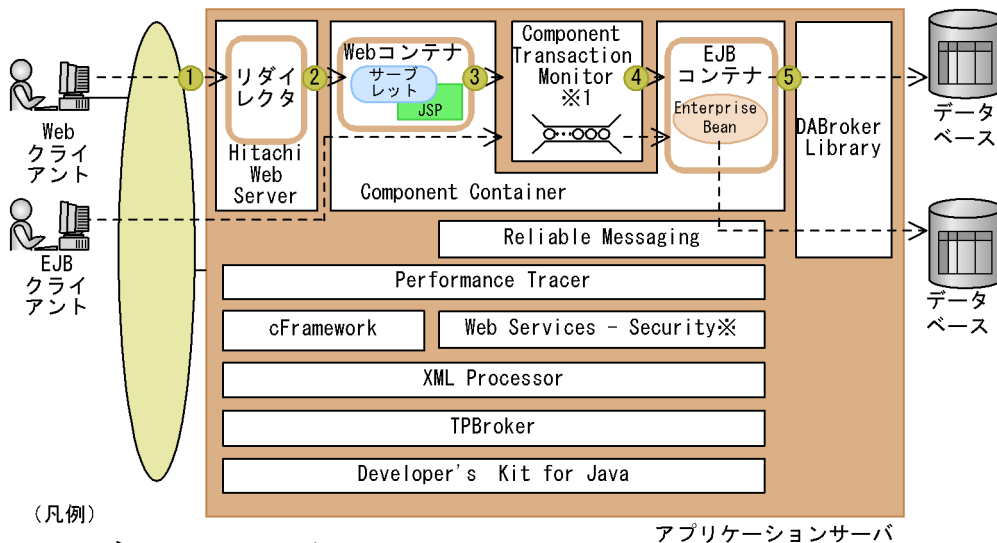
(1) J2EE アプリケーション実行時の構成ソフトウェアの位置づけ

アプリケーションサーバ上で J2EE アプリケーションを実行する場合に使用される Application Server の構成ソフトウェアの位置づけについて説明します。

Web クライアントまたは EJB クライアントからのリクエストを処理する場合に使用される Application Server の構成ソフトウェアの位置づけを次の図に示します。

2. Cosminexus の製品構成

図 2-8 Application Server の構成ソフトウェアの位置づけ (Web クライアントまたは EJB クライアントからのリクエストを処理する場合)



注1 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。

注2 図中の番号は、「(a)Webクライアントの場合」の説明に対応しています。

注※ Application Server Enterpriseの場合だけに含まれる構成ソフトウェアです。

参考

このほか、Application Server Enterprise では、Cosminexus Component Transaction Monitor の CTM のゲートウェイ機能を利用して TPBroker / OTM クライアントからのリクエストを受け付けることもできます。

構成ソフトウェアの位置づけとクライアントからのリクエストの流れについて、Web クライアントと EJB クライアントの場合に分けて説明します。

(a) Web クライアントの場合

リクエストの流れについて説明します。なお、項番は、図 2-8 の図中の番号に対応しています。

1. Web クライアントの Web ブラウザから送信したリクエストは、Web サーバ (Hitachi Web Server) が受け付けます。
2. Web サーバに登録されているリダイレクタというライブラリによって、リクエストは Web コンテナに振り分けられます。Web コンテナは、Cosminexus Component Container の構成要素の一つです。
Web コンテナでは、サブレットと JSP で構成される Web アプリケーションが動作しています。Web アプリケーションからは、業務処理プログラムである Enterprise

Bean を呼び出すことができます。

3. Application Server Enterprise の場合は、Web アプリケーションから業務処理プログラム (Stateless Session Bean) を呼び出すとき、Cosminexus Component Transaction Monitor (CTM) を使用してスケジューリングをするかどうかで、処理の流れが異なります。CTM によってリクエストのスケジューリングをする場合、Web アプリケーションから送信されたリクエストは、CTM のスケジュールキューに格納されます。CTM では、特定のサーバに負荷が集中しないように、キューを使用してリクエストの負荷分散や流量制御を実行します。
4. 適切にスケジューリングされたリクエストは、業務処理プログラムを実行する EJB コンテナに送信されます。なお、EJB コンテナは、Cosminexus Component Container の構成要素の一つです。また、CTM を使用していない場合は、Web アプリケーションから直接 EJB コンテナにリクエストが送信されます。EJB コンテナでは、リクエストに従って業務処理プログラムが実行されます。
5. EJB コンテナでは、処理内容に応じて、データベースへのアクセスが発生します。Cosminexus では、Cosminexus Component Container が提供する JCA 機能を使用して、データベースなどのエンタープライズ情報システム (EIS) に接続します。データベースとの連携には、必要に応じて Cosminexus DABroker Library を使用します。

処理が完了したリクエストの処理結果は、EJB コンテナから、CTM、Web コンテナ、Web サーバを経て、Web クライアントに返却されます。

(b) EJB クライアントの場合

EJB クライアントから送信したリクエストは、直接、CTM (Application Server Enterprise の場合) または EJB コンテナが受け付けます。そのあとの処理の流れは、Web クライアントからの処理の流れと同じです。

(2) 運用管理操作実行時の構成ソフトウェアの位置づけ

運用管理者から送信された Application Server に対する運用操作に関連するリクエストは、運用管理機能 (Cosminexus Manager 機能) が受け付けて、処理を実行します。Application Server の運用管理機能は、Cosminexus Component Container の機能として提供されています。なお、運用管理者からのリクエストは、CUI、Eclipse プラグインまたは Web ブラウザから送信されます。

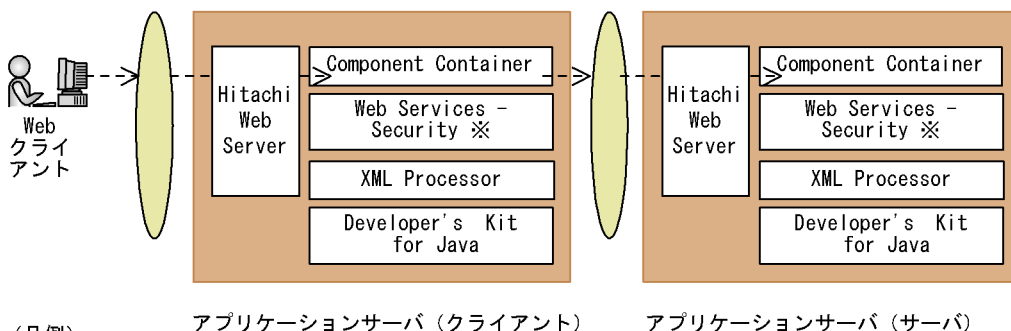
また、「(1) J2EE アプリケーション実行時の構成ソフトウェアの位置づけ」で説明したリクエスト処理の過程では、主要なポイントでトレース情報が出力されています。このトレース情報は、Cosminexus Performance Tracer によって収集され、ファイルに出力されています。運用管理者は、このトレース情報を収集、監視して、アプリケーションサーバ全体の処理性能とボトルネックの確認などができます。

(3) SOAP アプリケーション実行時の構成ソフトウェアの位置づけ

リクエストを実行するアプリケーションが SOAP アプリケーションの場合、Web クライアントなどから Web サーバが受け付けたリクエストは、SOAP のクライアントに当たるマシン内のライブラリ (SOAP クライアントライブラリ) を経由して、SOAP 通信によってサーバマシンに送信されます。サーバマシンでは、SOAP エンジンによってリクエストが処理されます。

SOAP アプリケーションを実行する場合に使用される Application Server の構成ソフトウェアの位置づけについて、次の図に示します。

図 2-9 Application Server の構成ソフトウェアの位置づけ (SOAP アプリケーションを実行する場合)



(凡例) -----> : リクエストの流れ

注 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。また、SOAP アプリケーションの実行 (SOAP 通信) に主に使用する構成ソフトウェアだけを記載しています。

注※ Application Server Enterprise の場合だけに含まれる構成ソフトウェアです。

SOAP アプリケーションを実行する場合、クライアントとサーバ間の SOAP 通信は、SOAP 通信基盤を介して実行されます。Cosminexus では、Cosminexus Component Container が SOAP 通信基盤として機能します。なお、SOAP クライアントライブラリおよび SOAP エンジンとしての機能も、Cosminexus Component Container が提供します。

また、SOAP 通信では、XML 形式のデータをやり取りします。このデータが盗聴や改ざんされることを防ぐために、Web サービスセキュリティの技術を利用できます。Web サービスセキュリティを利用するための機能は、Cosminexus Web Services - Security が提供しています。

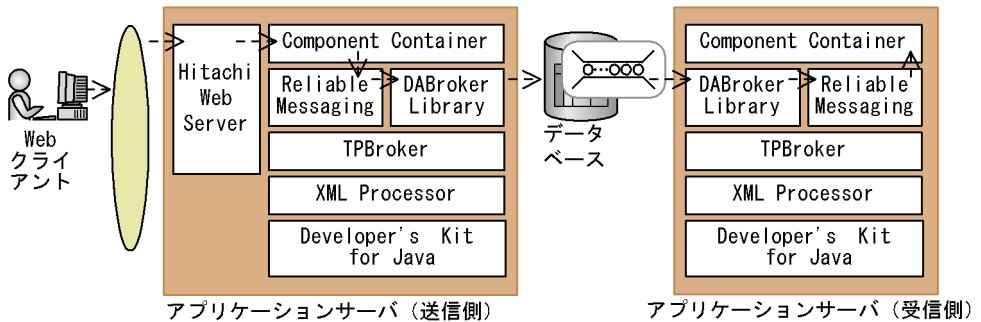
(4) メッセージによる処理の非同期実行時の構成ソフトウェアの位置づけ

アプリケーションサーバ間でメッセージの非同期通信を実行する場合、送信側のアプリケーションサーバが受け付けたメッセージは、キューに蓄積されます。受信側のアプリ

ケーションサーバがキューからメッセージを取り出したときに、メッセージの処理が実行されます。

メッセージによる非同期実行をする場合に使用される Application Server の構成ソフトウェアの位置づけを次の図に示します。

図 2-10 Application Server の構成ソフトウェアの位置づけ (メッセージによる処理の非同期実行をする場合)



(凡例)

-----> : リクエストの流れ

注 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。
また、メッセージによる処理の非同期実行で主に使用する構成ソフトウェアだけを記載しています。

アプリケーションの処理を非同期実行する場合、リクエストはメッセージとしてデータベース上のキューに格納されます。送信側でのメッセージのキューへの登録、および受信側でのキューからのメッセージの取り出しは、Cosminexus Reliable Messaging の機能で実行できます。キューを作成したデータベースへのアクセスには、Cosminexus Component Container の機能を使用します。また、Cosminexus DABroker Library は、JDBC ドライバとして必要に応じて使用します。

参考

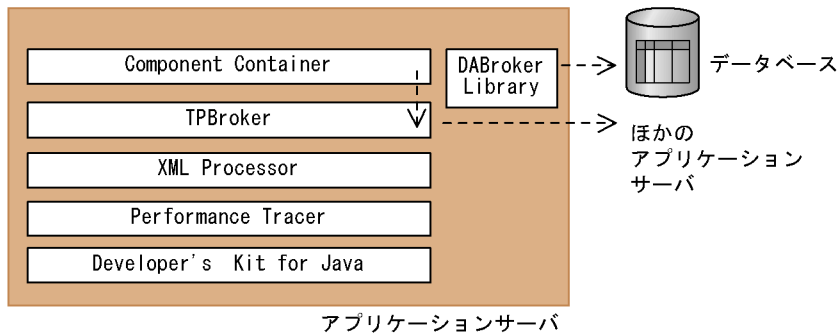
Cosminexus Reliable Messaging を使用するほか、Connector 1.5 仕様に準拠した任意のリソースアダプタを使用してメッセージによる処理の非同期実行を実現できます。任意のリソースアダプタを使用する場合の構成ソフトウェアおよびリソースアダプタの位置づけについては、使用するリソースアダプタの仕様をご確認ください。

(5) バッチアプリケーション実行時の構成ソフトウェアの位置づけ

アプリケーションサーバ上でバッチアプリケーションを実行する場合に、使用される Application Server の構成ソフトウェアの位置づけを次の図に示します。

2. Cosminexus の製品構成

図 2-11 Application Server の構成ソフトウェアの位置づけ (バッチサーバでバッチ処理を実行する場合)



(凡例)

----> : リクエストの流れ

注 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。

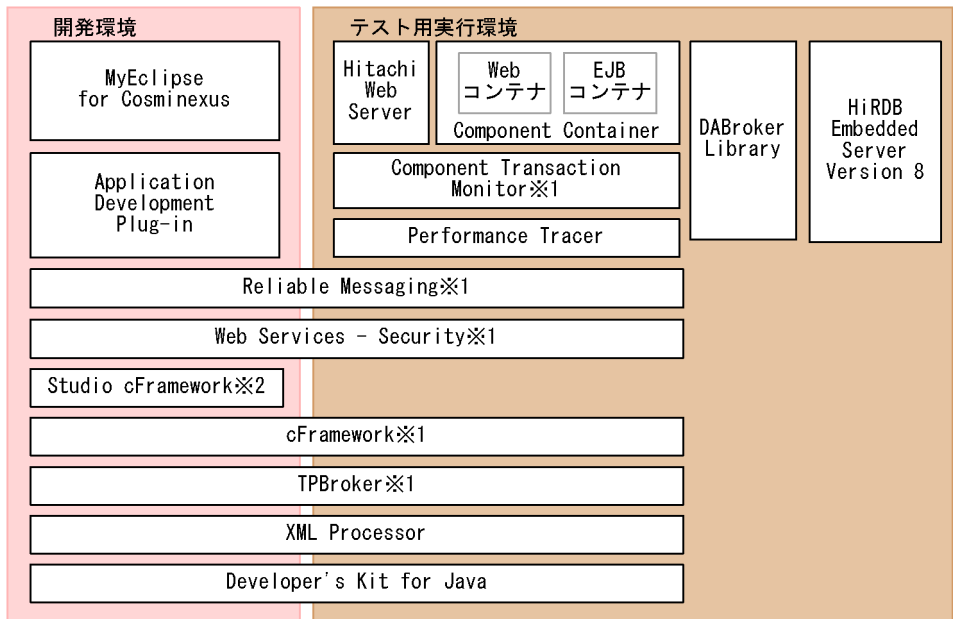
バッチアプリケーションは Cosminexus Component Container の機能を使用して実行されます。バッチアプリケーションからデータベースアクセスを実行する場合は、必要に応じて JDBC ドライバとして Cosminexus DABroker Library を使用します。また、ほかのアプリケーションサーバ上で動作している Enterprise Bean を呼び出す場合は、Cosminexus TPBroker の機能を使用してリモート呼び出しを実現します。

2.3.4 開発環境の構成ソフトウェアの位置づけ (Developer)

ここでは、開発環境を構築する製品である Developer (Developer Standard または Developer Professional) の各構成ソフトウェアの位置づけについて説明します。

Developer の構成ソフトウェアの位置づけを次の図に示します。

図 2-12 Developer の構成ソフトウェアの位置づけ



注 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。

注※1 Developer Professionalの場合だけに含まれる構成ソフトウェアです。

注※2 Developer Standardの場合だけに含まれる構成ソフトウェアです。

「開発環境」に含まれる構成ソフトウェアは、主にアプリケーション開発時に使用する構成ソフトウェアです。「テスト用実行環境」に含まれる構成ソフトウェアは、主にアプリケーションのテスト時に使用する構成ソフトウェアです。

(1) アプリケーション開発時に使用する構成ソフトウェアの位置づけ

Developer が提供する開発環境の構成ソフトウェアを使用して、J2EE アプリケーションを開発します。

MyEclipse for Cosminexus は、アプリケーションの統合開発環境です。Cosminexus Application Development Plug-in は、MyEclipse for Cosminexus からテスト環境上の J2EE サーバを操作するための機能（プラグイン）を提供します。

なお、アプリケーション開発時には、各構成ソフトウェアが提供する API を使用して J2EE アプリケーションを開発できます。

(2) アプリケーションのテスト時に使用する構成ソフトウェアの位置づけ

Developer が提供するテスト用実行環境の構成ソフトウェアを使用して、開発した J2EE アプリケーションを実行、テストし、デバッグをします。Developer 提供のプラグインを使用すると IDE からアプリケーションをデプロイできます。データベースを使用する

アプリケーションも、Developer で提供している HiRDB Embedded Server Version 8 を使用してテストできます。

2.3.5 サービス統合環境の構成ソフトウェアの位置づけ (Service Platform , Service Architect)

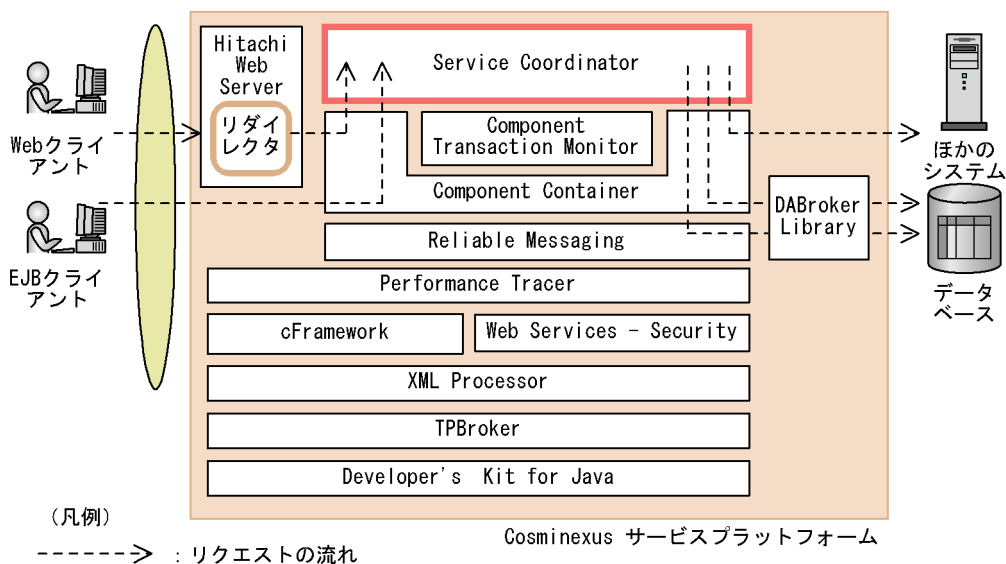
ここでは、サービス統合環境を構築する製品である Cosminexus サービスプラットフォーム (Service Platform , Service Architect) の各構成ソフトウェアの位置づけについて説明します。

(1) Service Platform の構成ソフトウェアの位置づけ

Service Platform では、Application Server Enterprise に加えて、サービス統合環境を構築するために必要な構成ソフトウェア Cosminexus Service Coordinator が追加されています。

Service Platform の構成ソフトウェアの位置づけを次の図に示します。

図 2-13 Service Platform の構成ソフトウェアの位置づけ



注 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。

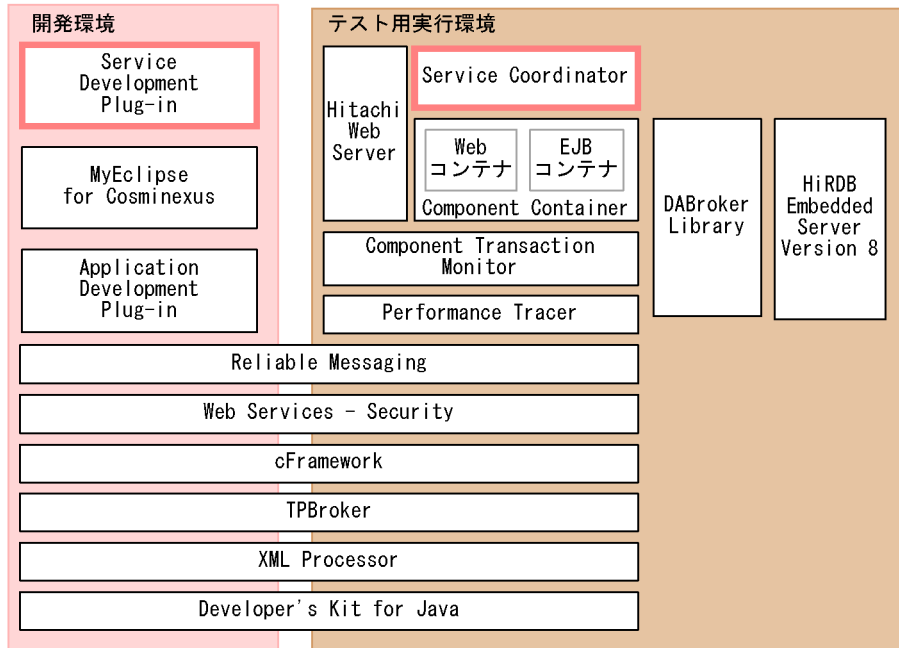
クライアントからのリクエストは、Cosminexus Component Container 経由で Cosminexus Service Coordinator が受け付けます。サービスの内容によって、同期処理の場合は SOAP プロトコルや RMI-IIOP プロトコルを利用してほかのシステム上のサービスを利用したり、非同期処理の場合は Cosminexus Reliable Messaging 経由でデータベース上のキューにリクエストを転送したりします。

(2) Service Architect の構成ソフトウェアの位置づけ

Service Architect では、Developer Professional に加えて、サービス統合環境を利用するために必要な構成ソフトウェアである、Cosminexus Service Development Plug-in および Cosminexus Service Coordinator が追加されています。

Service Architect の構成ソフトウェアの位置づけを次の図に示します。

図 2-14 Service Architect の構成ソフトウェアの位置づけ



注 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。

開発環境でサービス統合に必要な各種定義などを実行するためのプラグインの機能は、Cosminexus Service Development Plug-in によって提供される機能です。また、テスト用実行環境で Cosminexus サービスプラットフォームを構築するために、Cosminexus Service Coordinator を使用します。

2.4 動作環境

Cosminexus 製品の動作環境について説明します。

(1) 前提 OS

対象製品ごとに、前提 OS を示します。

(a) Application Server Standard , Application Server Enterprise および uCosminexus Web Redirector の前提 OS

Application Server Standard , Application Server Enterprise および uCosminexus Web Redirector の前提 OS について、次の表に示します。

表 2-4 Application Server Standard , Application Server Enterprise および uCosminexus Web Redirector の前提 OS

前提 OS	前提 OS のバージョン
Windows	Windows Server 2003 Windows Server 2003 (x64) Windows Server 2003 R2 Windows Server 2003 R2 (x64)
AIX	AIX 5L V5.2 AIX 5L V5.3
HP-UX	HP-UX 11i V2 (IPF) HP-UX 11i V3 (IPF)
Linux	Red Hat Enterprise Linux AS 3 (AMD64 & Intel EM64T) Red Hat Enterprise Linux AS 3 (IPF) Red Hat Enterprise Linux AS 3 (x86) Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T) Red Hat Enterprise Linux AS 4 (IPF) Red Hat Enterprise Linux AS 4 (x86) Red Hat Enterprise Linux 5 Advanced Platform (Intel Itanium) Red Hat Enterprise Linux ES 3 (AMD64 & Intel EM64T) Red Hat Enterprise Linux ES 3 (x86) Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) Red Hat Enterprise Linux ES 4 (x86)
Solaris	Solaris 9 Solaris 10

(b) Service Platform の前提 OS

Service Platform の前提 OS について、次の表に示します。

表 2-5 Service Platform の前提 OS

前提 OS	前提 OS のバージョン
Windows	Windows Server 2003 Windows Server 2003 (x64) Windows Server 2003 R2 Windows Server 2003 R2 (x64)
AIX	AIX 5L V5.2 AIX 5L V5.3
Linux	Red Hat Enterprise Linux AS 3 (AMD64 & Intel EM64T) Red Hat Enterprise Linux AS 3 (x86) Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T) Red Hat Enterprise Linux AS 4 (x86) Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64) Red Hat Enterprise Linux 5 Advanced Platform (x86) Red Hat Enterprise Linux ES 3 (AMD64 & Intel EM64T) Red Hat Enterprise Linux ES 3 (x86) Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) Red Hat Enterprise Linux ES 4 (x86) Red Hat Enterprise Linux 5 (AMD/Intel 64) Red Hat Enterprise Linux 5 (x86)
Solaris	Solaris 9 Solaris 10

(c) Developer Standard , Developer Professional , Service Architect および
uCosminexus Operator の前提 OS

Developer Standard , Developer Professional , Service Architect および uCosminexus Operator の前提 OS について , 次の表に示します。なお , これらの製品は , UNIX には対応していません。

表 2-6 Developer Standard , Developer Professional , Service Architect および
uCosminexus Operator の前提 OS

前提 OS	前提 OS のバージョン
Windows	Windows Server 2003 Windows Server 2003 R2 Windows Vista Business Windows Vista Enterprise Windows Vista Ultimate Windows XP

(d) uCosminexus Client の前提 OS

uCosminexus Client の前提 OS について , 次の表に示します。なお , uCosminexus Client は , UNIX には対応していません。

2. Cosminexus の製品構成

表 2-7 uCosminexus Client の前提 OS

前提 OS	前提 OS のバージョン
Windows	Windows Server 2003 Windows Server 2003 (x64) Windows Server 2003 R2 Windows Server 2003 R2 (x64) Windows Vista Business Windows Vista Enterprise Windows Vista Ultimate Windows XP

(2) Web 環境

前提となる Web 環境について示します。なお、インプロセス HTTP サーバを使用する場合、Web サーバは不要です。

(a) Web サーバ

前提となる Web サーバについて次の表に示します。

表 2-8 Cosminexus の前提となる Web サーバ

前提 OS	前提製品
Windows	Microsoft IIS 6.0 Hitachi Web Server
UNIX	Hitachi Web Server

注 Cosminexus 製品の構成ソフトウェアです。

(b) Web ブラウザ

前提となる Web ブラウザについて、次の表に示します。

表 2-9 Cosminexus の前提となる Web ブラウザ

Web 環境	前提製品
Web ブラウザ	HTTP1.0 または HTTP1.1 に対応したブラウザ (Internet Explorer 6.0 以降、または Mozilla FireFox1.0.72 以降)

(3) 言語

前提となる言語は、Java です。

(4) データベース

前提となるデータベースについては、「8.1 データベースとの連携」を参照してください。

3

システムのライフサイクル

この章では、Cosminexus で実現するシステムのライフサイクル、およびライフサイクルを構成する段階について説明します。また、それぞれの段階での作業の詳細について知りたい場合に参照するマニュアルを紹介します。

なお、この章では、Cosminexus をアプリケーションの実行環境および開発環境として使用する場合と、Cosminexus をサービス統合環境として使用する場合に分けて説明します。

3.1 実行環境および開発環境として使用する場合のシステムのライフサイクル

3.2 サービス統合環境として使用する場合のライフサイクル

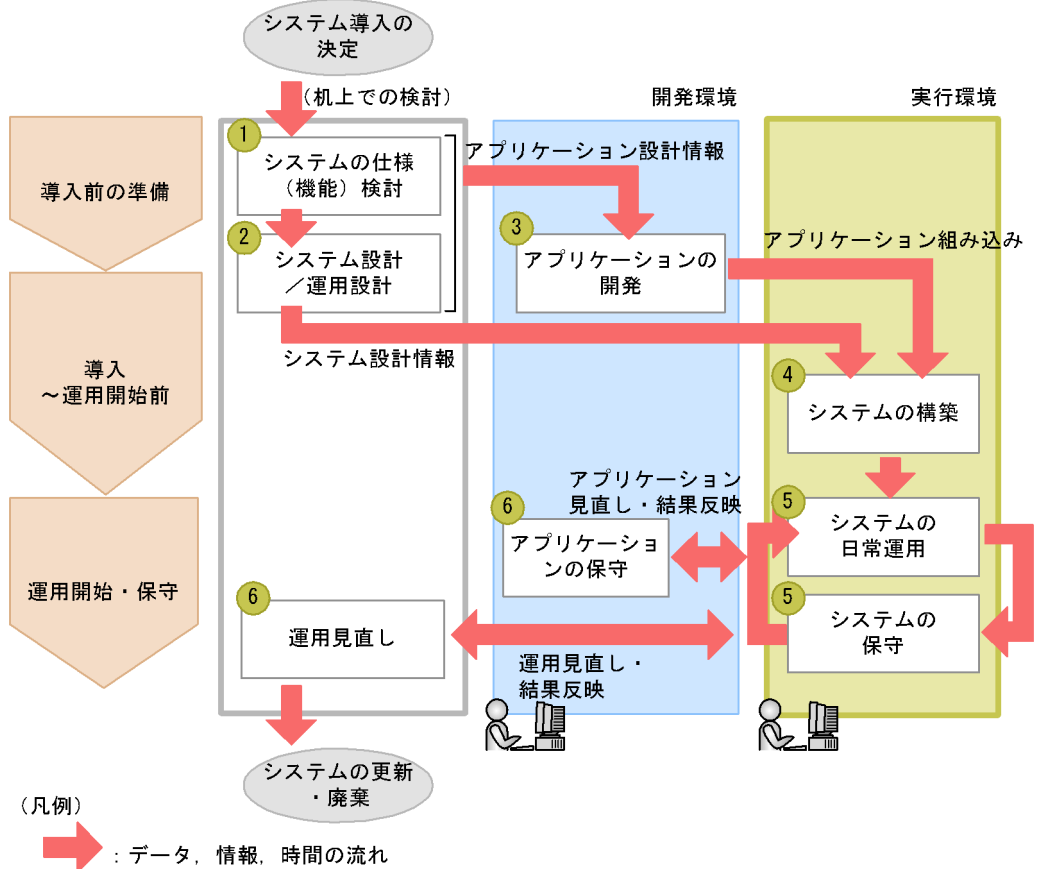
3.1 実行環境および開発環境として使用する場合のシステムのライフサイクル

実行環境および開発環境として使用する場合のシステムのライフサイクルには、次の段階があります。

1. システムの仕様（機能）検討
2. システム設計 / 運用設計
3. アプリケーションの開発
4. システムの構築
5. システムの日常運用と保守
6. 運用見直しとアプリケーションの保守

アプリケーションの開発環境および実行環境でのシステムのライフサイクルを図に表すと、次のようになります。

図 3-1 システムのライフサイクル



それぞれの段階について説明します。

1. システムの仕様（機能）検討
 システムの目的や業務の内容に応じて、Cosminexus のどの機能を使用するかを検討・決定します。具体的な作業内容については、「3.1.1 システムの仕様（機能）検討」で説明します。
2. システム設計 / 運用設計
 システムの目的や実行する業務、およびシステムの使用方法に応じてシステムの構成と運用方法を設計します。具体的な作業内容については、「3.1.2 システム設計と運用設計」で説明します。
3. アプリケーションの開発
 1. および 2. で検討したアプリケーション設計情報に基づいて、業務に合ったアプリケーションを開発します。具体的な作業内容については、「3.1.3 アプリケーションの開発」で説明します。
4. システムの構築

3. システムのライフサイクル

1. および 2. で検討したシステム設計情報に基づいて、システムを構築します。また、
3. で開発したアプリケーションをシステムに組み込みます。具体的な作業内容については、「3.1.4 システムの構築」で説明します。
5. システムの日常運用と保守
アプリケーションを実行して、システムの運用を開始します。システムの運用では、日常運用と定期的なシステム保守を繰り返します。具体的な作業内容については、「3.1.5 システムの運用と保守」で説明します。
6. 運用見直し / アプリケーション保守
業務内容の変更やシステム規模の変更などに合わせて、運用方法を随時見直します。また、必要に応じてアプリケーションも保守します。具体的な作業内容については、「3.1.5 システムの運用と保守」で説明します。

ライフサイクルは、システムの更新または廃棄を決定するまで続きます。

3.1.1 システムの仕様（機能）検討

システムの目的や業務の内容に応じて、実行環境でどの機能を使用するかを検討・決定します。

使用する機能の検討では、機能について説明しているマニュアルを必要に応じて参照してください。使用できる機能は、J2EE アプリケーションを実行するシステムを構築するか、バッチアプリケーションを実行するシステムを構築するかによって異なります。

J2EE アプリケーションを実行するシステムの場合に使用できる主な機能および参照先のマニュアルについて次の表に示します。

表 3-1 Cosminexus の主な機能および参照先のマニュアル（J2EE アプリケーションを実行するシステムの場合）

機能分類	概要	参照先
Java 言語	Java SE 5.0 に対応しています。	1
Web サーバ	Apache 2.0 に対応した Web サーバ（Hitachi Web Server）を使用できます。	Hitachi Web Server
Web コンテナ	Servlet 2.4 および JSP 2.0 に対応した Web コンテナを使用できます。また、インプロセス HTTP サーバ機能など、独自の機能を提供しています。	Cosminexus 機能解説
EJB コンテナ	次の Enterprise Bean を実行できる EJB コンテナを提供しています。 <ul style="list-style-type: none">• Session Bean• Entity Bean• Message-driven Bean また、Timer Service の機能も使用できます。	Cosminexus 機能解説

機能分類	概要	参照先
リソース接続とトランザクション管理	次の機能に対応したリソース接続とトランザクション管理ができます。 <ul style="list-style-type: none"> • コネクションプーリング • コネクションシェアリング • JNDI によるルックアップ • ローカルトランザクション • 2 フェーズコミットメント • 分散トランザクションの実現 	Cosminexus 機能解説
XML プロセッサ	次の機能を使用できます。 <ul style="list-style-type: none"> • DOM API • SAX API • XSLT トランスフォーマ (XPath を含む) 	Cosminexus XML Processor ユーザーズガイド
Web サービスおよび SOAP 間通信	次の機能を使用できます。 <ul style="list-style-type: none"> • RPC を利用した SOAP アプリケーションの開発 • メッセージングを利用した SOAP アプリケーションの開発 • EJB を利用した SOAP アプリケーションの開発 • UDDI クライアントライブラリが提供する JAXR API 	Cosminexus SOAP アプリケーション開発ガイド
OLTP 技術の適用	次の機能を使用できます。 <ul style="list-style-type: none"> • Web アプリケーションおよび URL グループ (業務ロジック) 単位の同時実行スレッド数制御 • Enterprise Bean の同時実行スレッド数制御 (CTM の利用) ² 	Cosminexus 機能解説
可用性向上	次の機能を使用できます。 <ul style="list-style-type: none"> • J2EE サーバ間のセッション情報の引き継ぎ • リソース枯渇監視 • 稼働情報監視 • 性能解析トレース / 障害解析トレースの出力 	Cosminexus 機能解説
セキュリティ管理	次の機能を使用できます。 <ul style="list-style-type: none"> • 統合ユーザ管理 • 監査ログ出力 • データベース監査証跡連携 	Cosminexus 機能解説
セキュリティ管理 (Web サービス) ²	次の機能を使用できます。 <ul style="list-style-type: none"> • SOAP メッセージの完全性および秘匿性の保証 • SOAP メッセージの認証 • XML 署名データの生成および検証 • XML 署名データの暗号化および復号化 	Cosminexus Web サービスセキュリティ 使用の手引 Cosminexus XML Security - Core ユーザーズガイド
フレームワークとライブラリ	次の機能を使用できます。 <ul style="list-style-type: none"> • JSF および JSTL • Cosminexus 提供のはん用部品 • cFramework の利用 	¹
アプリケーション開発	次の機能を使用できます。 <ul style="list-style-type: none"> • MyEclipse を使用したアプリケーションの開発 • ローカルマシンでのデバッグおよびリモートマシンでのデバッグ 	Cosminexus アプリケーション開発ガイド

3. システムのライフサイクル

機能分類	概要	参照先
システム構築	次の機能を使用できます。 <ul style="list-style-type: none"> システム構成の一括定義と簡易構築（Smart Composer 機能またはセットアップウィザード） リモート環境からの各種サーバの設定 アーカイブ形式または展開ディレクトリ形式でのアプリケーションのデプロイ リソースアダプタのデプロイ アプリケーションの設定 リソースの設定 複数実行環境の切り替え（UNIX の場合） 実行環境の移行性向上（構築済みのシステムの移行） 	Cosminexus 機能解説
システム運用	次の機能を使用できます。 <ul style="list-style-type: none"> サーバプロセスの一括起動、個別起動、自動起動、監視、再起動 J2EE アプリケーションのリロード JSP 事前コンパイル 稼働情報監視によるイベント発行と自動アクション制御 ログ/トレース収集 ドメイン一括管理 	Cosminexus 機能解説
クラスタソフトウェアとの連携	アプリケーションサーバまたは運用管理サーバを対象にした、コールドスタンバイでの 1:1 の系切り替えができます。また、相互スタンバイ構成や 1 台のリカバリ専用サーバを用意した N:1 リカバリシステム構成も実現できます。	Cosminexus 機能解説
JP1 との連携	JP1 の各製品と連携して次の機能を使用できます。 <ul style="list-style-type: none"> 障害の集中監視 稼働性能の監視 システム構成定義と管理 ジョブによる運用の自動化 シナリオによる運用の自動化 SNMP での稼働情報の取得 監査ログの収集と一元管理 	Cosminexus 機能解説

注 1 対応するマニュアルはありません。

注 2 Application Server Standard では使用できません。

バッチアプリケーションを実行するシステムの場合に使用できる主な機能および参照先のマニュアルについて次の表に示します。

表 3-2 Cosminexus の主な機能および参照先のマニュアル（バッチアプリケーションを実行するシステムの場合）

機能分類	概要	参照先
Java 言語	Java SE 5.0 に対応しています。	-

機能分類	概要	参照先
バッチサーバ	次の機能を実現できるバッチサーバを提供しています。 <ul style="list-style-type: none"> バッチ処理を実装した Java アプリケーションの実行 フルガーベージコレクション実行制御 	Cosminexus 機能解説
リソース接続とトランザクション管理	次の機能に対応したリソース接続とトランザクション管理ができます。 <ul style="list-style-type: none"> コネクションプーリング コネクションシェアリング JNDI によるルックアップ ローカルトランザクション 	Cosminexus 機能解説
XML プロセッサ	次の機能を使用できます。 <ul style="list-style-type: none"> DOM API SAX API XSLT トランスフォーマ (XPath を含む) 	Cosminexus XML Processor ユーザーズガイド
可用性向上	次の機能を使用できます。 <ul style="list-style-type: none"> リソース枯渇監視 稼働情報監視 性能解析トレース / 障害解析トレースの出力 	Cosminexus 機能解説
セキュリティ管理	次の機能を使用できます。 <ul style="list-style-type: none"> 監査ログ出力 データベース監査証跡連携 	Cosminexus 機能解説
アプリケーション開発	次の機能を使用できます。 <ul style="list-style-type: none"> MyEclipse を使用したアプリケーションの開発 ローカルマシンでのデバッグおよびリモートマシンでのデバッグ 	Cosminexus アプリケーション開発ガイド
システム構築	次の機能を使用できます。 <ul style="list-style-type: none"> システム構成の一括定義と簡易構築 (Smart Composer 機能) リモート環境からの各種サーバの設定 リソースアダプタのデプロイ リソースの設定 複数実行環境の切り替え (UNIX の場合) 実行環境の移行性向上 (構築済みのシステムの移行) 	Cosminexus 機能解説
システム運用	次の機能を使用できます。 <ul style="list-style-type: none"> サーバプロセスの一括起動, 個別起動, 自動起動, 監視, 再起動 稼働情報監視によるイベント発行と自動アクション制御 ログ/トレース収集 ドメイン一括管理 	Cosminexus 機能解説
クラスタソフトウェアとの連携	アプリケーションサーバを対象にした, コールドスタンバイでの 1:1 の系切り替えができます。また, 相互スタンバイ構成も実現できます。	Cosminexus 機能解説

3. システムのライフサイクル

機能分類	概要	参照先
JP1 との連携	JP1 の各製品と連携して次の機能を使用できます。 <ul style="list-style-type: none">• 障害の集中監視• 稼働性能の監視• システム構成定義と管理• ジョブによる運用の自動化• シナリオによる運用の自動化• SNMP での稼働情報の取得• 監査ログの収集と一元管理	Cosminexus 機能解説

注 バッチアプリケーションを実行するシステムは、Application Server Enterprise で構築できません。

注 対応するマニュアルはありません。

3.1.2 システム設計と運用設計

Cosminexus のシステム設計と運用設計では、次の点を検討します。

システム構成をどうするか/システムのスケラビリティをどのように確保するかまず、J2EE アプリケーションによってオンライン処理を実行するシステムか、バッチアプリケーションによってバッチ処理を実行するシステムかを明確にします。次に、システムの規模に応じて、Web サーバ、J2EE サーバ、バッチサーバおよび各プロセスを物理的にどのように配置するかを決めます。オンライン処理を実行するシステムの場合は、複数のサーバを使用して負荷を分散する必要があるかなどを検討します。スケラビリティについてもよく検討しておく必要があります。実際の日常運用が開始されたあとで、システム規模を変更したり、トラブル発生時の影響をできるだけ局所的に抑えたりするためには、システムのスケラビリティが確保されていることが重要です。

どのような運用のしかたをするか/可用性と信頼性をどのように確保するか業務システムを運用する場合には、システムの可用性と信頼性を確保しながら、できるだけ効率良く運用していく方法を検討します。Cosminexus には、システム内の複数のサーバマシンを一括して管理、運用するための機能として、Management Server という運用管理機能があります。

また、システム全体を効率良く運用する方法として、JP1 やクラスタソフトウェアと連携するかどうか、あわせて検討してください。

システムのセキュリティをどのように確保するかミッション・クリティカルな業務システムの場合、信頼性の高いセキュリティを確保することは不可欠です。システム設計の段階で、セキュアなシステムを構築するための観点を明確にして、導入・運用時の手順を決めておく必要があります。また、外部ネットワークと接続するシステムを構築する場合は、ファイアウォール、侵入検知システムおよび SSL アクセラレータを適切に配置、設定してどのように信頼性の高いシステムを構築するかを、あわせて検討してください。

性能を向上させるためにどのようなチューニングが必要か

システム要件によっては、厳密な性能設計が必要です。プールやキャッシュを効果的に利用したり、タイムアウトを適切に設定したりすることで、システム全体の性能向上が図れます。

JavaVM をどのようにチューニングするか

アプリケーションサーバで動作する J2EE サーバなどのプロセスは、JavaVM 上で実行されます。JavaVM で使用するメモリ空間を適切に管理することによって、フルガーベージコレクションの頻発を防ぎ、パフォーマンスの低下を防げます。

システム設計および運用設計の方法の詳細については、マニュアル「Cosminexus システム設計ガイド」を参照してください。

3.1.3 アプリケーションの開発

目的とする業務内容に合わせて、アプリケーションを開発します。

アプリケーションサーバでは、Java 言語を使用して開発した J2EE アプリケーションやバッチアプリケーションを実行できます。

一般的な J2EE アプリケーションは、MVC アーキテクチャに基づいて、コンポーネント化されたプログラム群によって構成されます。個々の業務処理プログラムは、EJB の仕様に従った Enterprise Bean として作成します。Enterprise Bean の組み合わせによって複雑なアプリケーションを開発したり、業務の変化に応じて Enterprise Bean を入れ替えたりすることで、再利用性の高い J2EE アプリケーションが開発できます。

Cosminexus システムで動作するアプリケーションは、次のような方法で開発できます。

Cosminexus が提供する開発環境 (Developer) の機能を使用して開発する Developer を使用する場合、Eclipse のプラグインである MyEclipse を使用してアプリケーションを開発できます。

MyEclipse では、UML を使用したアプリケーションの設計から、コーディング、ビルド、デバッグまでの一連の開発作業を実行できます。

また、構築したテスト環境上の J2EE サーバの起動・停止や、アプリケーションの開始・停止などの操作も、MyEclipse から実行できます。

Cosminexus の開発環境および IDE を使用しないで開発する

Developer や IDE を使用しなくても、アプリケーションは開発できます。この場合は、テキストエディタを使用してプログラムのソースを作成し、javac コマンドでコンパイルします。また、jar コマンドでアーカイブを作成して、アプリケーションを開発します。

アプリケーションの開発方法の詳細については、マニュアル「Cosminexus アプリケーション開発ガイド」を参照してください。

3.1.4 システムの構築

システム設計の結果を基に、システムを構築します。

システム構築とは、Cosminexus の実行環境を、J2EE アプリケーションまたはバッチアプリケーションが実行できる状態にすることです。J2EE アプリケーションを実行するシステムを構築する場合は、Application Server のインストール、J2EE サーバの設定、Web サーバとの連携、J2EE リソースと J2EE アプリケーションの設定、運用環境の構築などが含まれます。バッチアプリケーションを実行するシステムを構築する場合は、Application Server のインストール、バッチサーバの設定、リソースの設定などが含まれます。

Cosminexus のシステムは、Smart Composer 機能が提供しているコマンドを使用して構築します。Smart Composer 機能のコマンドを使用すると、J2EE サーバ、バッチサーバ、Web サーバなどのサーバプロセスを、一括で構築できます。また、J2EE アプリケーションを実行するシステムを構築する場合は、ウィザードプログラムによってシステムを構築する機能である、セットアップウィザードを使用することもできます。

Smart Composer 機能やセットアップウィザードを使用して構築したシステムは、Management Server というサーバプロセスの機能を使用して一括管理できます。

システム構築の概要と構築手順については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。また、Cosminexus で使用できる機能ごとの詳細な設定方法については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

3.1.5 システムの運用と保守

アプリケーションの開発とシステムの構築ができれば、アプリケーションを実行して、運用を開始します。システムの運用では、日常の運用とシステム保守を繰り返します。

日常運用では、日常的なサーバの起動/停止のほか、システムを安定稼働させるために各種プロセスの監視、ログの収集、ユーザの管理などをします。

システムの保守では、スケールイン、スケールアウト、スケールアップ、スケールダウンなどのシステム規模の変更をしたり、発生したトラブルに対処（トラブルシューティング）したりします。

Smart Composer 機能を使用して構築したシステムでは、複数のホスト、プロセスおよびアプリケーションを、一括操作および一括監視できます。一括操作および一括監視には、Management Server の運用管理コマンド、または Smart Composer 機能のコマンドが使用できます。

システムの運用方法のうち、システムの起動、停止およびステータスの監視などの作業の詳細については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。それ以外の運用作業の詳細については、マニュアル「Cosminexus システム運用

ガイド」を参照してください。

なお、運用方法について見直す場合は、必要に応じてシステム構築からやり直してください。

また、さらに広範囲なシステム全体をまとめて統合運用するために、JP1 を使用することもできます。JP1 と連携して実現できる運用については、マニュアル「Cosminexus 機能解説」の JP1 と連携したシステムの運用の説明を参照してください。それぞれの JP1 製品の詳細な機能および設定手順については、使用する JP1 製品のマニュアルを参照してください。

3.2 サービス統合環境として使用する場合のライフサイクル

サービス統合環境として使用する場合のシステムのライフサイクルには、次の段階があります。

1. システム設計 / サービスの準備
2. 環境の構築およびセットアップ
3. 各種定義 / アプリケーションの作成
4. システムの運用
5. システムの保守と見直し

それぞれの段階で実施する内容については、マニュアル「Cosminexus サービスプラットフォーム 概説」を参照してください。

ここでは、サービス統合環境で独自の段階である、サービスの準備、および各種定義 / アプリケーションの作成について説明します。

3.2.1 サービスの準備

サービスの準備では、システムで実行する業務を分析し、業務をどのようにサービス化するかを検討します。既存のシステムで提供されているサービスを再利用したり、新しくアプリケーションを開発してサービスにしたりすることを検討します。

3.2.2 各種定義 / アプリケーションの作成

クライアントから Cosminexus サービスプラットフォームを介して各サービスを利用するために、Cosminexus サービスプラットフォームの開発環境で必要な定義を実施します。複数のサービスを組み合わせる一つのサービスとして利用するための定義（ビジネスプロセスの定義）も、開発環境で実施します。また、任意の Java プログラム開発環境で、サービスを利用するためのリクエストを送信するクライアントアプリケーションを開発します。

4

J2EE アプリケーションを実行する

この章では、J2EE アプリケーションを実行するための環境の概要について説明します。J2EE アプリケーションの実行環境は、Application Server Standard、Application Server Enterprise または Service Platform を使用して構築・運用します。

なお、開発環境用の製品である、Developer Standard、Developer Professional および Service Architect で構築したテスト用実行環境でも、この章で説明する機能を使用できます。

4.1	J2EE アプリケーションの実行環境の特長
4.2	J2EE アプリケーションの実行環境の種類
4.3	J2EE アプリケーションの実行環境を構築・運用する
4.4	Web サービスに対応したアプリケーションの実行環境を構築・運用する
4.5	メッセージを利用した非同期通信アプリケーションの実行環境を構築・運用する
4.6	導入から運用までの流れ

4.1 J2EE アプリケーションの実行環境の特長

この節では、J2EE アプリケーションの実行環境の特長について説明します。

4.1.1 標準仕様への対応

J2EE の標準仕様に準拠したアプリケーションサーバでは、J2EE サーバの機能によって、J2EE の API を使用した J2EE アプリケーションを実行できます。Cosminexus は、J2EE 1.4 に準拠しています。J2EE アプリケーションとは、サーブレット、JSP で構成される Web アプリケーションや Enterprise Bean で構成される J2EE アプリケーションです。

アプリケーションサーバ上では、業務ごとに開発された J2EE アプリケーションが実行されます。J2EE に準拠した実行環境で動作する J2EE アプリケーションでは、複数の業務で共通に使用される機能を、J2EE API を使用して実現できます。共通に使用される機能とは、例えば、データベースやメインフレームに接続する機能や、セッション管理機能、トランザクション管理機能などです。これらを API で実現できることで、個々の J2EE アプリケーション内での煩雑なコーディングを減らすことができます。これらの機能は、アプリケーションサーバに含まれるモジュールである、コンテナ、サービスなどの形式で、アプリケーションに提供されます。

このほか、Cosminexus で構築したアプリケーションサーバは、XML、Web サービス、SSL、分散オブジェクトなどに関連した標準仕様にも対応しています。Cosminexus が対応する標準仕様については、「付録 A Cosminexus が対応する標準仕様」を参照してください。

4.1.2 システムの安定稼働の実現

業務システムには、安定して稼働し続けることが求められます。業務の内容によっては、一定時間にアクセスが急増したり、特定の処理に要求が集中したりする場合があります。急激な状況の変化に即応して、システムを安定稼働させるためには、次の制御が有効です。

流量制御

クライアントからのアクセス要求や処理要求などのリクエストに対して、処理の同時実行数を制御することで、リクエスト増加時のパフォーマンス低下を防いでシステムを安定稼働させる方法です。

優先制御

クライアントからのリクエスト増加時に、特定の処理要求を優先して処理することで、処理の緊急度や優先度に応じたリクエスト処理を実現する方法です。

負荷分散

リクエストを処理するサーバを 1 か所に集中させないで、複数の実行環境で分散させ

ることで、パフォーマンスの低下を防いでシステムを安定稼働させる方法です。

Cosminexus では、J2EE アプリケーションに対して、きめ細やかな流量制御、優先制御を実現できます。これによって、システムの安定稼働に加えて、システムリソースの有効活用も実現できます。

ここでは、Web アプリケーションの流量制御、および OLTP 技術を適用した Enterprise Bean の流量制御と負荷分散について説明します。なお、それぞれの機能の詳細については、マニュアル「Cosminexus 機能解説」を参照してください。

(1) Web アプリケーションの流量制御

アプリケーションサーバでは、サーブレットや JSP によって構成される Web アプリケーションを、J2EE サーバの Web アプリケーション実行基盤である Web コンテナで実行します。

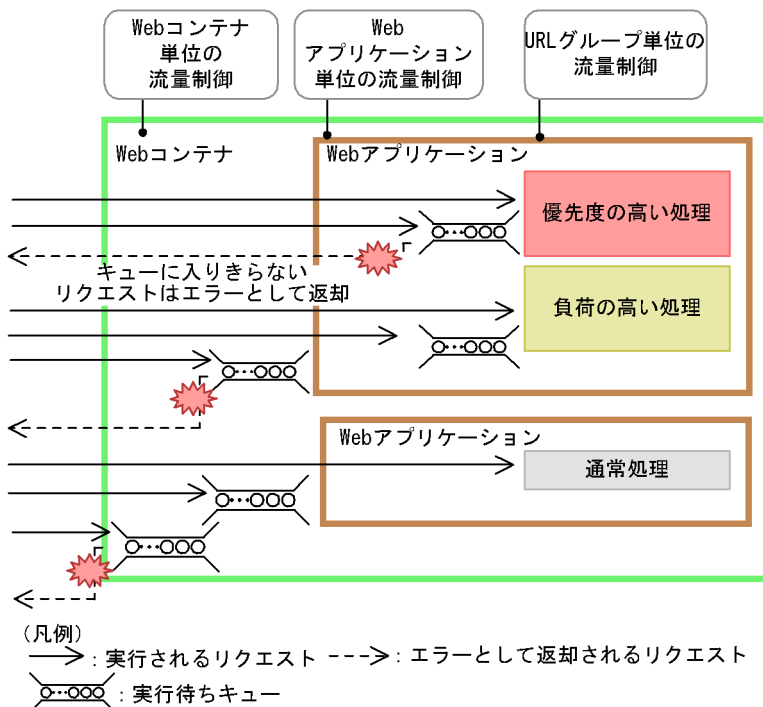
Cosminexus の Web コンテナでは、Web コンテナ単位、Web アプリケーション単位および Web アプリケーション内の URL グループ（業務ロジック）単位に同時に処理できるリクエスト数を設定したり、リクエストをキュー（待ち行列）の概念で管理したりできます。これによって、Web アプリケーションで同時に実行する処理数を、処理内容に応じて細かく制御できます。送信されたリクエスト数が突発的に急増した場合でも、処理するリクエスト数を一定の数に制御できるので、システムを安定した状態で稼働させることができます。

また、リクエストの重要度に応じたリクエストの実行が可能になります。例えば、緊急度の高い重要な処理は確実に処理できるようにすることで、負荷の高い業務処理がほかの業務に影響を与えることを防げます。

Web アプリケーションの流量制御の概要を次の図に示します。

4. J2EE アプリケーションを実行する

図 4-1 Web アプリケーションの流量制御の概要



Web コンテナ単位、Web アプリケーション単位および URL グループ（業務ロジック）単位で流量制御を実施することによって、システムの安定稼働を実現しながら、負荷の高い処理があっても優先度の高い処理を確実に実行できるシステムを構築できます。

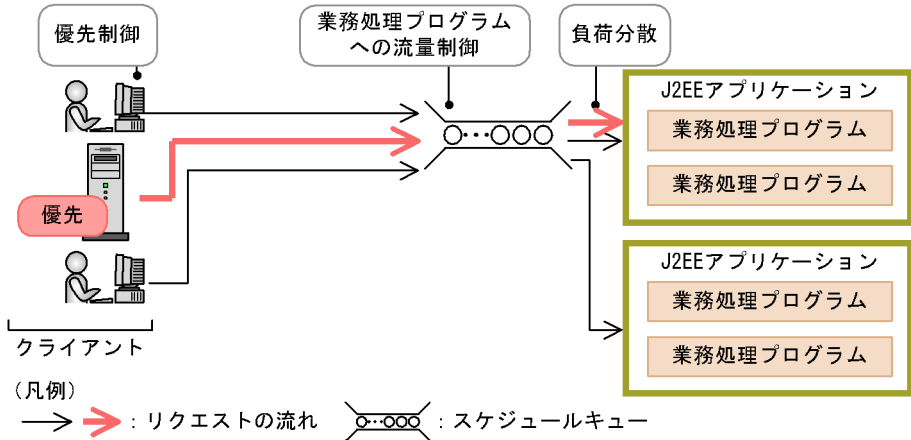
(2) OLTP 技術の Enterprise Bean への適用

OLTP 技術は、大規模な業務システムには不可欠な技術です。Cosminexus では、J2EE アプリケーション内の業務処理プログラム（Enterprise Bean）に対して、高度な OLTP 技術を適用できます。これによって、次のような処理を実現できます。なお、対象となる Enterprise Bean は、Stateless Session Bean です。

リクエストのスケジューリングを利用した流量制御と負荷分散

業務処理プログラムに対するリクエストをスケジューリングすることで、特定の J2EE アプリケーションに大量のリクエストが集中した場合に、リクエストを複数のサーバに振り分けて処理させたり、一度に処理させるリクエストの数（流量）を制御したりできます。また、リクエストの送信元であるクライアントに優先順位を設定して、そのクライアントから送信されたリクエストを優先的に処理するようにできます。さらに、特定の業務処理プログラムに対する処理を適切に分散させて負荷の集中を防ぐことで、システム全体としての処理性能の向上と、システムの安定稼働を図れます。Enterprise Bean の優先制御、流量制御および負荷分散の概要を次の図に示します。

図 4-2 Enterprise Bean の優先制御，流量制御および負荷分散の概要



J2EE アプリケーション内の業務処理プログラムへのリクエストを，スケジュールキューを経由させて実行することで，一度に大量のリクエストが送信された場合も同時に実行する数を制御したり，負荷分散したりできます。また，優先制御によって，重要なリクエストを速やかに確実に実行できます。

サービス閉塞の実現

J2EE アプリケーションの稼働中に特定の業務処理プログラムを入れ替えたい場合に，対象の業務処理プログラムに対応するリクエストを制御して，関連するサービスだけを安全に閉塞できます。特定の業務処理プログラムに障害が発生した場合などに，該当箇所を局所化し，縮退運転と回復によって，システム全体を止めないで業務処理プログラムの入れ替えができます。

4.1.3 可用性と耐障害性の向上

ミッションクリティカルな業務システムには，システムが提供するサービスをできるだけ停止することなく，安定して提供し続けられる仕組みが求められます。

例えば，提供するサービスの内容によっては，24 時間連続稼働が求められるサービスや，障害が発生して業務システムが止まることで大きな損失が発生するサービスもあります。

このため，業務システムの基盤であるアプリケーションサーバには，まず，障害を未然に防ぐこと，そして，もし障害が発生しても，障害の影響範囲を局所的に抑え，業務システムを止めることなく運用し続けられることが求められます。また，障害が発生した個所は迅速に回復できることが必要です。

ここでは，Cosminexus で実現できる，可用性と耐障害性の高いシステムの特長について説明します。なお，それぞれの機能の詳細については，マニュアル「Cosminexus 機能解説」を参照してください。

4. J2EE アプリケーションを実行する

(1) 障害発生の未然防止

アプリケーションやリソースの稼働状態を監視して、障害の予兆を検知し、障害発生を予防できます。また、しきい値を設定して監視することで、しきい値を超えた場合に自動的に対処するためのアクションも定義できます。

例えば、リクエストの集中によって JavaVM でフルガーベージコレクションが多発する場合に、一定時間内に発生するフルガーベージコレクションの回数にしきい値を設定して監視できます。しきい値を超えた場合にリクエストの同時実行数を動的に減らすなどの処理を自動実行することで、システムのスローダウンを予防できます。

(2) 障害発生時の可用性向上

障害発生時の可用性を高めるためには、次の機能を使用できます。

きめ細やかなタイムアウトの設定

クライアントと Web サーバ間、Web サーバと Web コンテナ間、EJB クライアントと EJB コンテナ間、EJB クライアントとネーミングサービス間などできめ細かくタイムアウトが設定できます。これによって、通信先のマシンに障害が発生している場合でも、数秒で障害を検知でき、無応答などによってサービスを停滞させません。また、アプリケーション内のメソッドレベルの処理にもタイムアウトを設定できます。これによって、無限ループなどの問題がアプリケーション内で発生した場合に速やかにその処理をキャンセルして、業務を継続できます。

クラスタソフトウェアとの連携

クラスタソフトウェアと連携してアプリケーションサーバや運用管理用のサーバをクラスタ構成にすることで、障害発生時に速やかに系切り替えを実行し、サービスの停止を防げます (1:1 クラスタ構成)。

また、リソース使用中のアプリケーションサーバで障害が発生した場合に備え、1 台の待機サーバを用意しておくことで、障害発生時にすぐにリソースを解放しトランザクション処理を決着できます (N:1 クラスタ構成)。

セッション情報の引き継ぎ

Web システムで、すでにクライアントからのセッションが確立されているサーバに障害が発生したときに、セッション情報をほかのサーバに引き継いで、サービスを継続できます。

バックエンドシステムとの再接続

データベースなどのバックエンドシステムとの接続が異常終了した場合、自動再接続できます。

(3) 業務のコンポーネント化

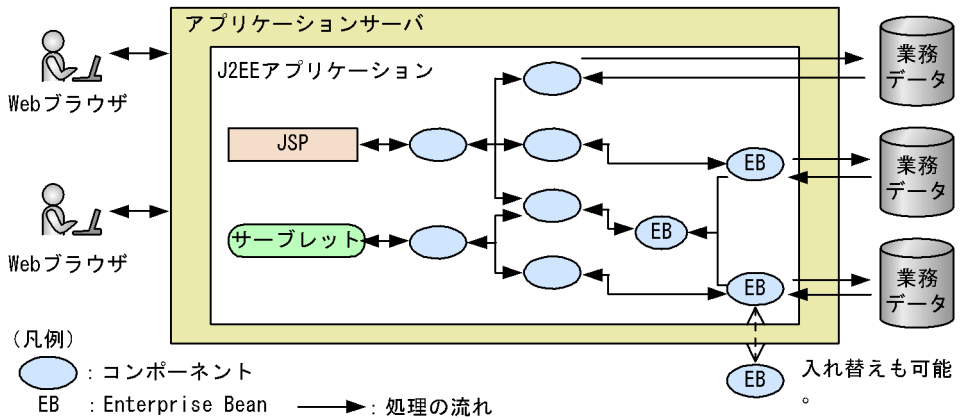
業務をコンポーネント化することで、可用性の高いシステムが構築できます。

Cosminexus では、サーバサイドのコンポーネント技術である、EJB の仕様に対応したコンテナおよびサービスを提供することで、J2EE に準拠したアプリケーションサーバを構築できます。

EJB の仕様では、それぞれの業務処理プログラムが、Enterprise Bean としてコンポーネント化されます。Enterprise Bean は、J2EE アプリケーションに組み込まれて実行されます。業務処理が変更になった場合に、該当する Enterprise Bean だけを入れ替えることもできます。

業務のコンポーネント化の概要を次の図に示します。

図 4-3 業務のコンポーネント化の概要



4.1.4 システム導入および拡張の容易化

業務システムは、Web サーバ、J2EE サーバ、負荷分散機、運用管理用のサーバなど、複数の要素で構成されています。また、J2EE サーバ内にも、複数の機能があります。アプリケーションサーバを中心とした業務システムを、特長を生かして円滑に運用するためには、システムの用途に応じたシステム構成を検討し、それぞれの要素の多様なパラメタを、適切な関係で設定する必要があります。

ここでは、システムを導入および拡張するときの、Cosminexus の特長について説明します。機能の詳細については、マニュアル「Cosminexus 機能解説」を参照してください。

(1) システムの一括構築の実現とスケールアウトの容易化

J2EE アプリケーションを実行するシステムは、次のどちらかの方法で一括して構築できます。

定義ファイルにパラメタを定義してコマンドを実行する方法

対話形式の CUI プログラムの画面に設定値を入力する方法

構築するシステムの構成によって、これらの方法を使い分けることができます。

また、同じ業務サービスを提供するシステムは、ユニットという概念で管理できます。ユニット単位でシステム構成の変更やスケールアウトができるため、サービス提供規模

4. J2EE アプリケーションを実行する

などの変更に対応して、スムーズにシステムを拡張できます。

(2) テスト環境から本番環境へのスムーズな移行

システムを導入する場合、本番稼働用の実行環境（本番環境）でシステムを構築する前に、テスト用の実行環境（テスト環境）を構築して、テストおよびチューニングを実施することがあります。

Cosminexus では、テスト環境でチューニング済みの定義をまとめてファイルに出力できます。本番環境を構築するときには、このファイルを本番環境用の定義として使用できます。これによって、テスト環境でチューニングした内容を再度本番環境で設定する必要がなくなり、構築済みのシステムのスムーズな移行を実現できます。

(3) 複数の構築済み実行環境の切り替えが可能

本番稼働している実行環境のアプリケーションサーバに対して、バージョンアップやパッチの適用が必要な場合があります。この場合、業務に支障が出ないように、作業中または作業後にトラブルが発生したときに速やかに元の実行環境を復元できるようにしておく必要があります。

Cosminexus では、バージョンアップまたはパッチ適用の前の実行環境についての情報を、一括して退避しておくことができます。退避後には、バージョンアップまたはパッチ適用後の別の実行環境を、同じマシン内に構築できます。なお、問題が発生した場合には、退避しておいた環境を速やかに復元できます。

これによって、複数の構築済み実行環境の切り替えが、効率良く、かつ確実に実行できます。

なお、この機能は、UNIX の場合に使用できる機能です。

4.1.5 システム監査によるシステムのセキュリティ確保

会計不祥事などの問題が多く発生する中で、組織には、内部統制の強化が強く求められています。

内部統制の目的は、いつ、だれが、どんな業務を実行したかを把握して、業務が各種法規制に準拠して遂行されていることを検証することです。このため、内部統制に対応するには、次のことが求められます。

- 正しい権利を持つ担当者が、正しい操作によって適切に業務を実施したことを検証できること。
- 検証結果に問題がないことを、監査者や評価者に対して証明できること。

これらに対応するためには、業務システムで、「だれが」「いつ」「何を実施したか」を監査し、監査結果を記録として管理しておく必要があります。

Cosminexus では、次の機能を提供しています。

- 監査ログの出力

- データベースと連携した監査証跡情報の出力

ここでは、これらの機能の概要を説明します。それぞれの機能の詳細については、マニュアル「Cosminexus 機能解説」を参照してください。

(1) 監査ログの出力

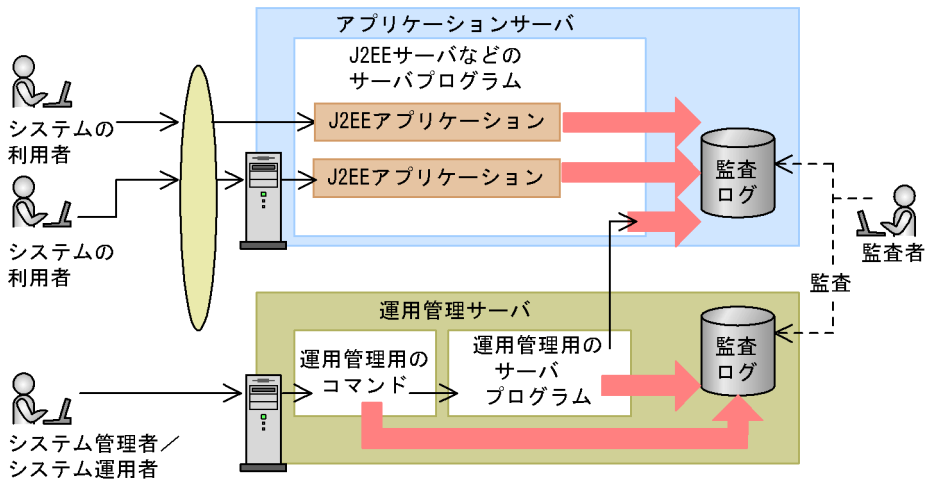
Cosminexus で構築した業務システムでは、「だれが」「いつ」「何を実施したか」についての情報を、監査ログとして出力できます。

監査ログに出力されるのは、次の情報です。

- システム管理者やシステム運用者が実施した操作の履歴と、それに伴うプログラムの動作の履歴
- J2EE アプリケーションを通してシステムの利用者が実行した操作の履歴と、それに伴うプログラムの動作の履歴

Cosminexus での監査ログ出力の概要を次の図に示します。

図 4-4 Cosminexus での監査ログ出力の概要



(凡例)

→ : アプリケーション実行またはコマンド実行による制御の流れ

---> : 監査の流れ

➡ : 監査ログ出力の流れ

なお、Cosminexus が出力した監査ログは、JP1 と連携して、Cosminexus 以外の日立のミドルウェアが出力する監査ログとまとめて管理できます。

JP1 との連携については、「8. ほかの製品との連携」を参照してください。

4. J2EE アプリケーションを実行する

(2) データベースと連携した監査証跡情報の出力

Cosminexus で構築した業務システムのバックエンドでは、多くの場合、データベースが動作しています。データベースには、漏洩や改ざんが許されない重要なデータが多く格納されています。これらの情報は、適正なセキュリティ管理が行われ、厳重に管理される必要があります。

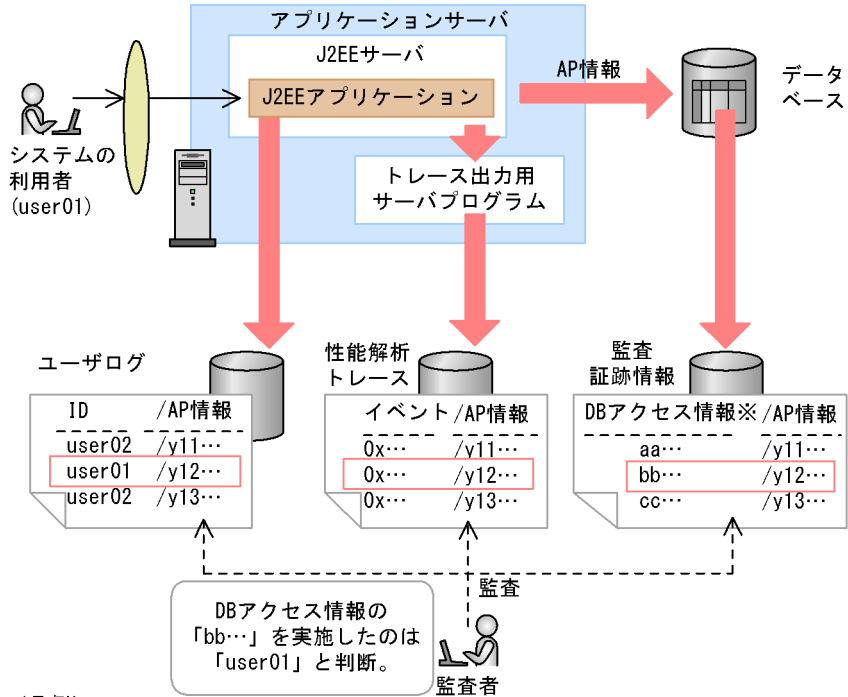
データベースには、「だれが」「いつ」「どのようなデータベースアクセスを実行したか」を示す情報を出力できるものがあります。この情報を、監査証跡情報といいます。

Cosminexus では、データベースが出力する監査証跡情報に、アプリケーションサーバのどのリクエストでデータベースアクセスが実行されたのかを示す情報を出力できます。この情報と J2EE アプリケーションで出力するログ情報などを組み合わせると、データベースアクセスが、Cosminexus のどのユーザの操作の延長として実行されたのかを追跡できます。

なお、Cosminexus が監査証跡情報を出力するために連携できるデータベースは、HiRDB です。

データベースと連携した監査証跡情報の出力の概要を次の図に示します。

図 4-5 データベースと連携した監査証跡情報の出力の概要



注 AP情報は、ルートアプリケーション情報（リクエストを特定するための情報）を示します。

注※ データベースが管理しているデータベースの操作履歴が出力されます。

この例では、システムの利用者（user01）がJ2EEアプリケーションを経由してデータベースにアクセスするとき、次の3種類の情報が出力されています。

- J2EEアプリケーションで出力するユーザログ
- トレース出力用サーバプログラム（パフォーマンストレーサ）が出力する性能解析トレース
- データベースで出力する監査証跡情報

これらの情報には、すべてリクエストを特定するための情報（ルートアプリケーション情報）が出力されています。監査者は、この情報を利用して、データベースアクセスがどのリクエストの延長として実行されたのか、そのリクエストを実行したのはどのユーザなのか、などを検証します。また、性能解析トレースを使用すると、そのリクエストがどのような流れで処理されたのかを検証することもできます。

4.1.6 業務効率を向上させる運用管理の実現

業務システムの運用管理は、規模が大きくなるほど複雑になります。できるだけ効率良く、運用コストを抑え、かつシステムの性能を最大限に生かせる運用方法が求められま

4. J2EE アプリケーションを実行する

す。そのためには、操作性に優れたインタフェースや、自動運用の仕組みが必要です。

Cosminexus では、これらの要件を満たすために、次の運用管理機能を提供しています。なお、それぞれの機能の詳細については、マニュアル「Cosminexus 機能解説」を参照してください。

業務システムの一括運用

運用管理ドメインという概念で、複数の J2EE サーバや Web サーバなどの各種サーバを一括管理できます。また、JP1 と連携することで、Cosminexus 以外で構築されたシステムも含めた業務システム全体に対して、監視、運用の自動化、稼働状況の分析などの一括運用ができるようになります。

運用作業の自動化

障害発生時のサーバ再起動や、運用監視機能と連携したイベント発行・アクション実行などの機能を利用することで、障害発生時の回復作業の自動化や、障害予防作業の自動化を図れます。

また、JP1 と連携することで、Cosminexus 以外のシステムも含めたシステムの運用作業を自動化できます。

ログの運用

Cosminexus のシステムが出力したログを一括して収集できます。また、ユーザが開発した J2EE アプリケーションのログを Cosminexus のシステムが出力したログと同じように扱うことができます。システム全体のログを一括管理することで、信頼性の高いログ運用を実現できます。

また、Cosminexus で使用する日立の JavaVM では、障害発生時の要因分析やシステムの状態確認に利用できるよう、ログの出力内容が拡張されています。このログを利用して適切なチューニングを実施することで、システムの可用性向上を図れます。

トレース情報による性能解析

リクエストの実行時に、Web サーバからデータベース接続までの各ポイントでトレース情報を出力します。この情報を収集して分析することで、処理のボトルネックを明確にしたり、障害発生の原因になった個所を特定したりできます。

統合ユーザ管理

Cosminexus で構築したシステムにログインするユーザを統合管理できます。それぞれの J2EE アプリケーションが管理しているユーザの情報が関連づけられて管理されるため、一度のログイン処理でさまざまな J2EE アプリケーションにログインできるようになります。なお、ユーザ情報を管理するリポジトリとしては、LDAP ディレクトリサービスやデータベースが利用できます。

メンテナンス時のアプリケーション入れ替えの効率化

アプリケーションをアーカイブ形式と展開ディレクトリ形式の 2 種類で管理できます。展開ディレクトリ形式のアプリケーションの場合、アプリケーションを入れ替えるときにアーカイブし直す必要がなく、変更したクラスファイルをリロードするだけで入れ替えができます。また、アーカイブ形式のアプリケーションの場合も、局所的な変更の場合にはアプリケーションの停止や再設定が不要な入れ替え（リデプロイ）がで

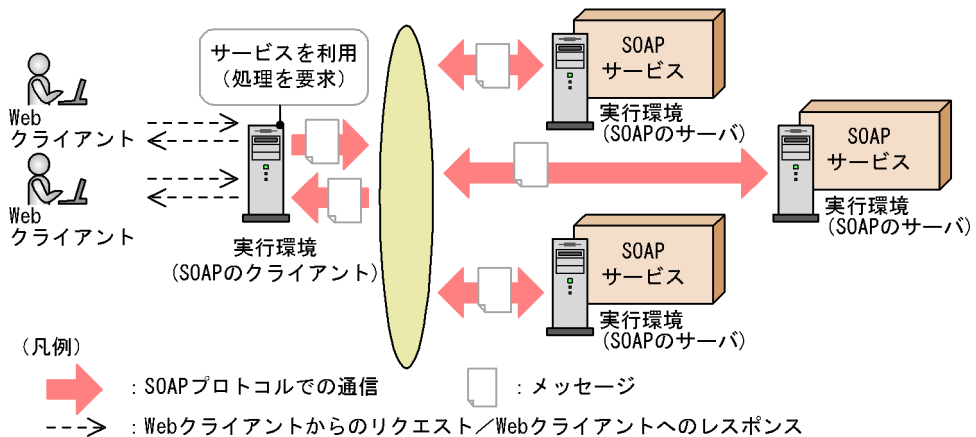
きます。これらによって、メンテナンス時の入れ替え効率を高め、システム運用コストを抑えられます。

4.1.7 Web サービスへの対応

Web サービスとは、システム間でプログラムを使用して情報をやり取りし、ほかのシステムが持つアプリケーションを、Web 上で使用できるようにした仕組みです。Web サービスの多くは、SOAP と呼ばれるプロトコルを使用して、ネットワーク上でのシステム間連携を実現します。SOAP では他システムのアプリケーションへのアクセスを実現するために、下位のトランスポート層に Web 標準の HTTP プロトコルを利用します。Cosminexus では、Web サービスの実行環境としての機能を提供するほか、Web サービスに対応したアプリケーションの開発を支援しています。

Web サービス利用環境の概要を次の図に示します。

図 4-6 Web サービス利用環境の概要



Web サービスの利用環境には、SOAP のクライアントになる実行環境と、SOAP のサーバになる実行環境が必要です。Cosminexus では、これら両方の実行環境を構築できます。

Web クライアントのユーザからのリクエストを受け付けた実行環境 (SOAP のクライアント) は、サービスを利用するためのリクエストを、SOAP サービスを提供する実行環境 (SOAP のサーバ) に送ります。実行環境 (SOAP のサーバ) では、SOAP サービスによって処理を実行して、結果をレスポンスとして実行環境 (SOAP のクライアント) に送信します。SOAP のクライアントと SOAP のサーバ間の通信には SOAP プロトコルを使用します。なお、Web サービスの利用環境で動作し、ネットワークを利用して公開、実行できるアプリケーションのことを SOAP アプリケーションといいます。

機能の詳細については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。

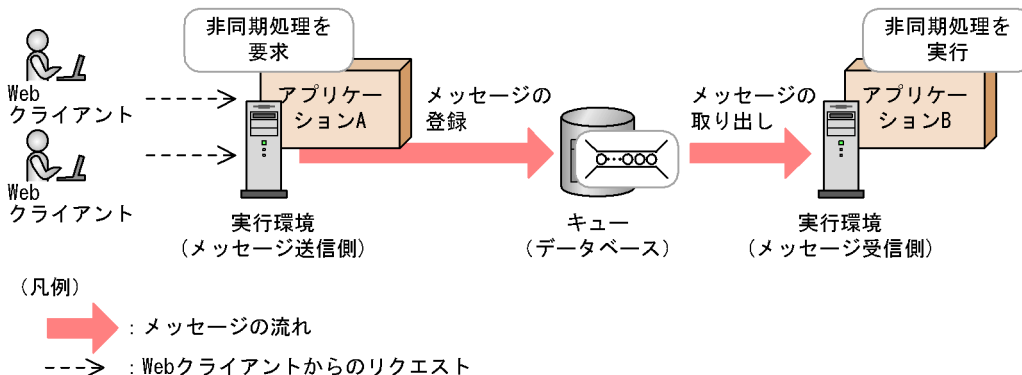
4.1.8 信頼性の高い非同期通信の実現

アプリケーション間で、メッセージを使用した非同期通信を実現できます。

非同期通信では、送信側のアプリケーションがメッセージを送信する際、キューにメッセージを登録します。受信側のアプリケーションは、メッセージを受信して、キューからメッセージを取り出します。これによって、送信側、受信側のアプリケーションが同時に動作していない場合も、メッセージのやり取りが可能になります。

Cosminexus で実現できる非同期通信の概要を次の図に示します。

図 4-7 Cosminexus で実現できる非同期通信の概要



メッセージを管理するキューをデータベースで管理するため、高信頼なメッセージ管理ができます。また、確実に1回の配送保証、または順序保証のQoS（通信品質）などによってメッセージ通信の信頼性も確保できます。さらに、JMS インタフェースを採用し、高信頼な標準プロトコルである WS-Reliability へも対応しているため、ほかのシステムとの連携やほかのシステムからのアプリケーションの移植などにも柔軟に対応できます。

機能の詳細については、マニュアル「Cosminexus Reliable Messaging」を参照してください。

参考

この項の説明は、Cosminexus の構成ソフトウェアである Cosminexus Reliable Messaging を使用した場合の非同期通信の説明です。

このほか、Connector 1.5 仕様に準拠した任意のリソースアダプタを使用して、外部のリソースからのメッセージ送信を受け付けてアプリケーションサーバ上の処理を非同期で実行することもできます。詳細は、マニュアル「Cosminexus 機能解説」を参照してください。

4.2 J2EE アプリケーションの実行環境の種類

Cosminexus では、J2EE に準拠した J2EE アプリケーションの実行環境を構築できます。また、クライアントとサーバ間の通信に SOAP を利用する SOAP アプリケーションの実行環境や、サーバ間の通信にメッセージを利用して非同期通信を実現するアプリケーションの実行環境も構築できます。

実行するアプリケーションの種類によって、実行環境の構成および使用する機能が異なります。例えば、J2EE アプリケーションでサーブレットや JSP を使用するか、または Enterprise Bean を使用するかによって、推奨されるシステム構成が異なります。さらに、システム構成によって、構築方法も異なります。

以降では、実行環境を次の 3 種類に分けて説明します。

1. J2EE アプリケーションの実行環境 (2. および 3. に該当しない実行環境)
「4.3 J2EE アプリケーションの実行環境を構築・運用する」で説明します。
2. Web サービスを利用するためのアプリケーション (SOAP アプリケーション) の実行環境
「4.4 Web サービスに対応したアプリケーションの実行環境を構築・運用する」で説明します。
3. サーバ間でメッセージを利用した非同期通信を実行するアプリケーションの実行環境
「4.5 メッセージを利用した非同期通信アプリケーションの実行環境を構築・運用する」で説明します。

なお、2. および 3. の環境は、1. の環境を前提としています。このため、これらの実行環境で利用する機能の検討やシステム構築、運用をする場合は、1. の実行環境で実施する内容に加えて 2. と 3. の環境で必要な内容を実施するようにしてください。

4.3 J2EE アプリケーションの実行環境を構築・運用する

この節では、J2EE アプリケーションの実行環境の構築・運用について説明します。

4.3.1 システム構成の種類

ここでは、実行環境のシステム構成として、次の2種類のシステム構成について説明します。

スケールアウトシステム

フロントエンド/バックエンドシステム

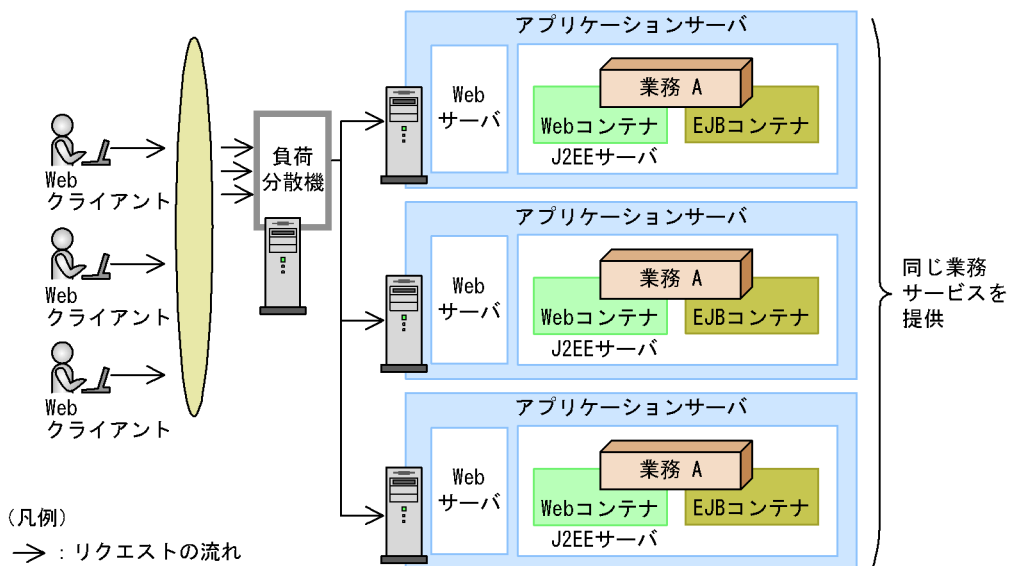
(1) スケールアウトシステム

スケールアウトとは、システム全体の処理性能を向上させることを目的として、サーバの台数を増やすことです。一方、サーバの台数を減らすことでシステム規模を縮小することをスケールインといいます。

スケールアウトシステムはシステム規模の拡大や縮小に柔軟に対応できる、スケラビリティの高いシステムのことで、業務システムの運用規模や要件の変化に応じてスケールアウトまたはスケールインを実施できます。

スケールアウトシステムの構成例を次に示します。

図 4-8 スケールアウトシステムの構成例



この図の構成の場合、3台のアプリケーションサーバは同じ業務を実行するサーバです。

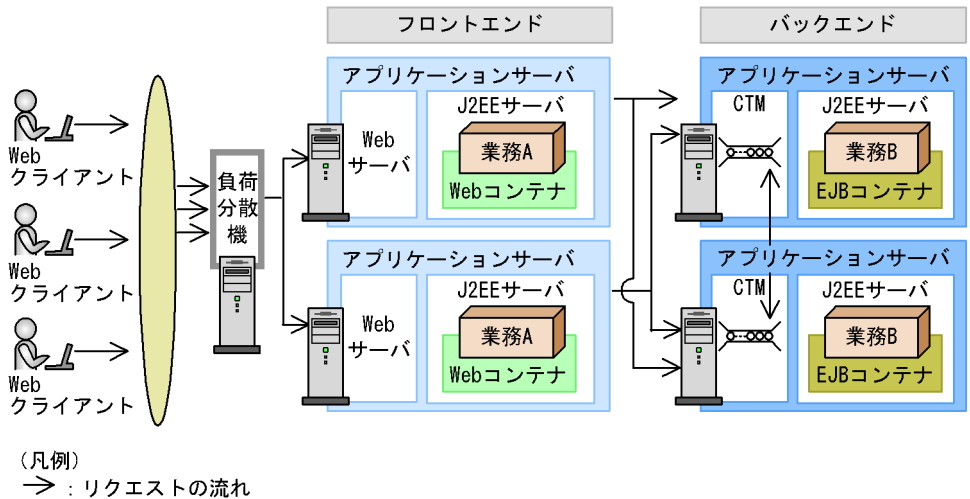
リクエスト数の増加などによって処理性能を上げる場合は、さらにアプリケーションサーバマシンの台数を増やすことで対応できます。

(2) フロントエンド/バックエンドシステム

フロントエンド/バックエンドシステムは、システムのスケラビリティを確保しながら可用性を高めたシステムです。フロントエンドとバックエンドから構成されます。フロントエンドでは、Web ブラウザなどからのリクエストを受け付けて適切な業務処理プログラムに処理を振り分ける処理を実行します。バックエンドでは、フロントエンドから振り分けられた業務処理を実行し、必要に応じて EIS にアクセスします。

アプリケーションサーバの場合、フロントエンドでサーブレットや JSP を実行して、バックエンドで Enterprise Bean を実行する構成などが該当します。フロントエンド/バックエンドシステムの構成例を次に示します。

図 4-9 フロントエンド/バックエンドシステムの構成例



この図の構成の場合、フロントエンドの2台のアプリケーションサーバでは同じ処理を実行します。また、バックエンドの2台のアプリケーションサーバでも同じ処理を実行します。クライアントからのリクエストは、まず、負荷分散機によってフロントエンドのアプリケーションサーバに振り分けられます。また、Enterprise Bean のスケジューリング機能を持つ CTM を使用することによって、バックエンドでの負荷分散も実現できます。

フロントエンド、バックエンドはそれぞれスケールアウト構成になっているため、リクエスト数の増加などによって処理性能を上げる場合は、各層のアプリケーションサーバの台数を増やすことで対応できます。また、障害発生時の縮退運用や再起動時の影響の局所化などが図れるため、システム全体の可用性を向上させることができます。

4. J2EE アプリケーションを実行する

ポイント

バックエンドで CTM を利用する場合、バックエンドのマシンに次のどれかの製品が必要です。

- Application Server Enterprise
- Service Platform

なお、アプリケーションの開発環境でのテスト時に CTM を利用する場合は、次のどれかの製品を使用できます。

- Developer Professional
 - Service Architect
-

4.3.2 スケールアウト構成のシステムの構築・運用

Cosminexus では、システムを一括構築・運用するための機能として、次の機能を提供しています。

Smart Composer 機能

複数台のマシンで構成される J2EE アプリケーションの実行環境を一括して構築・運用できます。構築には、定義ファイルやコマンドを使用します。

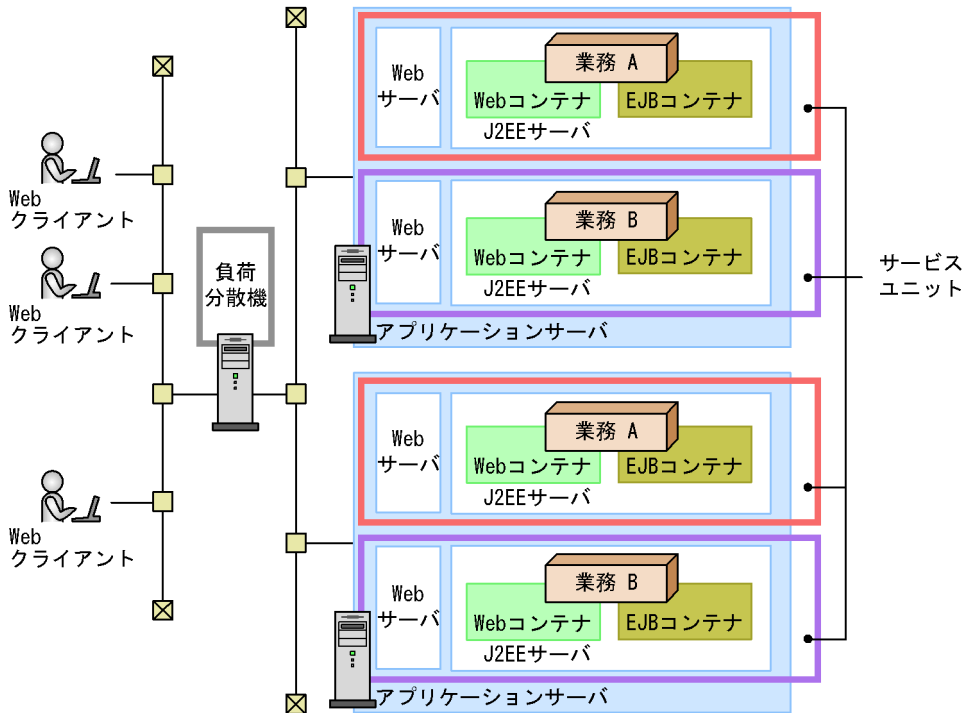
セットアップウィザード

1 台のマシンで構成される J2EE アプリケーションの実行環境を構築・運用できます。構築には、対話形式の CUI プログラムを使用します。

Cosminexus の推奨構成は、Web サーバと J2EE サーバを 1 : 1 の組み合わせで配置するスケールアウト構成です。なお、一つの業務処理を実行する Web サーバと J2EE サーバの組み合わせは、サービスユニットとして管理できます。サービスユニットは、一つの業務を実行するためのサーバの集合です。サービスユニット単位に、スケールアウトや、閉塞処理を実行できます。セッション情報を管理するためのサーバや、CTM を使用したリクエストのスケジューリングに使用するサーバの集合も、サービスユニットとして扱えます。

次に、スケールアウト構成の例を示します。この例では、一つのマシン内に二つのサービスユニットを配置しています。サービスユニットを二つ配置することで、例えば、業務 1 としてオンライン系の業務を実行し、業務 2 としてオフライン系の業務を実行するなどの運用も実現できます。

図 4-10 Smart Composer 機能によって構築するスケールアウト構成の例



4.3.3 フロントエンド/バックエンド構成のシステムの構築・運用

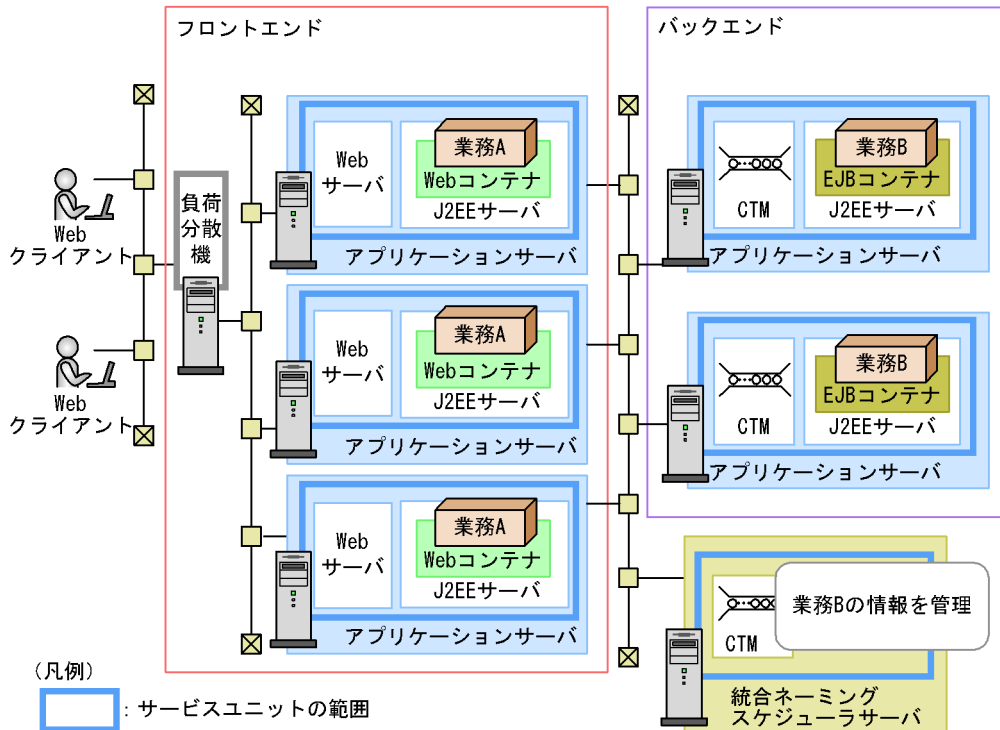
フロントエンド/バックエンドシステムを構築する場合も、Smart Composer 機能またはセットアップウィザードを使用して構築できます。この場合は、フロントエンドのサービスユニットとバックエンドのサービスユニットを、それぞれスケールアウト構成で構築します。ただし、セットアップウィザードは、1 台のマシンで構成されるシステムを構築する場合だけに使用できます。

なお、Smart Composer 機能で CTM を使用するバックエンドシステムを構築する場合、CTM のスケジューリングの対象になる業務処理プログラム (Enterprise Bean) の情報を登録したサーバ (統合ネーミングスケジューラサーバ) は、J2EE アプリケーションとは別のサービスユニットとして扱います。

CTM を使用したフロントエンド/バックエンド構成として構築した例を次の図に示します。

4. J2EE アプリケーションを実行する

図 4-11 Smart Composer 機能によって構築するフロントエンド/バックエンド構成の例 (CTM を使用する場合)



このほか、次のようなパターンの構成も、Smart Composer 機能を使用して構築できます。

負荷分散機の冗長化構成 (アクティブ - スタンバイ構成)

セッションの引き継ぎ (セッションフェイルオーバー機能) を実現する構成

Web サーバと J2EE サーバを分けた構成

インプロセス HTTP サーバ機能を使用する構成

Cosminexus で実現できるシステム構成については、マニュアル「Cosminexus システム設計ガイド」を参照してください。また、Smart Composer 機能またはセットアップウィザードを使用した構築・運用方法については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

4.4 Web サービスに対応したアプリケーションの実行環境を構築・運用する

この節では、Web サービスに対応したアプリケーションの実行環境について説明します。

4.4.1 SOAP アプリケーションの実行環境

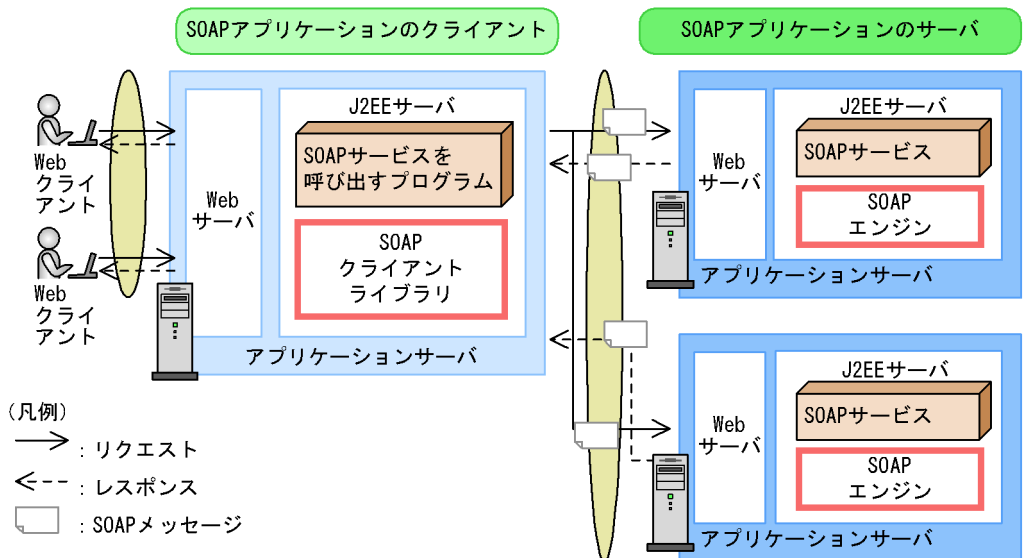
Web の標準技術を利用してネットワーク上でサービスを公開、実行できるアプリケーションのことを、SOAP アプリケーションといいます。

SOAP アプリケーションでは、クライアントとサーバ間のメッセージ交換用のプロトコルに SOAP を利用しています。クライアント側はサーバにメッセージを送信して、サーバ側で提供される SOAP サービスを利用します。なお、SOAP アプリケーションのクライアントとサーバ間でやり取りするメッセージを SOAP メッセージといいます。

SOAP アプリケーションの実行環境は、SOAP サービスを呼び出すクライアントと、SOAP サービスを提供するサーバで構成されます。クライアントには、SOAP サービスを呼び出すための SOAP クライアントライブラリが必要です。サーバには、クライアントと SOAP サービスの送受信の仲介をするエンジンである、SOAP エンジンが必要です。

SOAP アプリケーションの実行環境の例を次の図に示します。

図 4-12 SOAP アプリケーションの実行環境の例



SOAP アプリケーションのサーバは、アプリケーションサーバとして構築します。クライアントは、アプリケーションサーバ、またはコマンドラインで動作する Java アプリ

4. J2EE アプリケーションを実行する

ケーションの実行環境として構築します。

図 4-12 の場合は、クライアントをアプリケーションサーバとして構築しています。Web クライアントからのリクエストを受け付けたアプリケーションサーバが SOAP アプリケーションのクライアントになり、サーバに当たるアプリケーションサーバ上の SOAP サービスを呼び出します。SOAP アプリケーションのクライアントでは、SOAP クライアントライブラリを使用して、サーバにメッセージを送信します。メッセージを受け取ったサーバでは、SOAP エンジンを通じて SOAP サービスを実行し、レスポンスをメッセージとしてクライアントに返します。

ポイント

SOAP アプリケーションのクライアントとサーバに使用できる製品を次の表に示します。

表 4-1 SOAP アプリケーションのクライアントとサーバに使用できる製品（Web サービスセキュリティの技術を使用しない場合）

構築する実行環境	使用できる製品
SOAP アプリケーションのクライアント	Application Server Standard Application Server Enterprise Service Platform Developer Professional Service Architect uCosminexus Client
SOAP アプリケーションのサーバ	Application Server Standard Application Server Enterprise Service Platform Developer Professional Service Architect

注 開発環境でのテスト用実行環境として使用できます。

SOAP アプリケーションで実現できることの詳細については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。

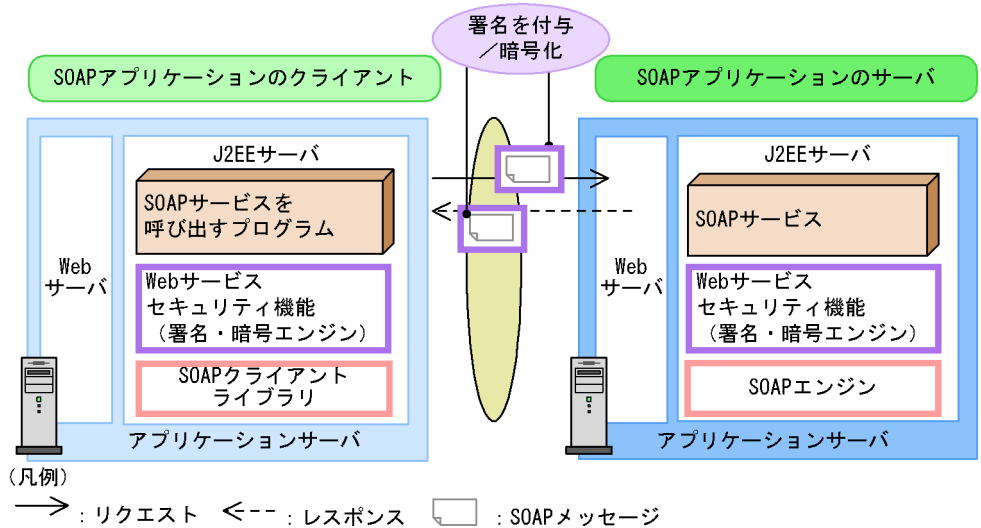
4.4.2 Web サービスセキュリティの技術を利用した SOAP アプリケーションの実行環境

SOAP アプリケーションのクライアントとサーバ間の通信で送受信する SOAP メッセージは、XML 形式のデータです。このデータの内容を改ざんされたり第三者による盗聴を防いだりするための技術として、Web サービスセキュリティという技術を使用できます。

Web サービスセキュリティの技術によって、SOAP メッセージの完全性と秘匿性が保証されます。また、SOAP メッセージの認証も実現できます。さらに、XML 署名データを生成または検証したり、データを暗号化または復号化したりすることもできます。

Web サービスセキュリティの技術を利用したシステムの例を次の図に示します。

図 4-13 Web サービスセキュリティの技術を利用したシステムの例



ポイント

Web サービスセキュリティの技術を使用する場合に、SOAP アプリケーションのクライアントとサーバに使用できる製品を次の表に示します。

表 4-2 SOAP アプリケーションのクライアントとサーバに使用できる製品 (Web サービスセキュリティの技術を使用する場合)

構築する実行環境	使用できる製品
SOAP アプリケーションのクライアント	Application Server Enterprise Service Platform Developer Professional Service Architect
SOAP アプリケーションのサーバ	Application Server Enterprise Service Platform Developer Professional Service Architect

注 開発環境でのテスト用実行環境として使用できます。

Web サービスセキュリティ技術の詳細については、マニュアル「Cosminexus Web サービスセキュリティ 使用の手引」を参照してください。XML 署名および XML 暗号の詳細については、マニュアル「Cosminexus XML Security - Core ユーザーズガイド」を参照してください。

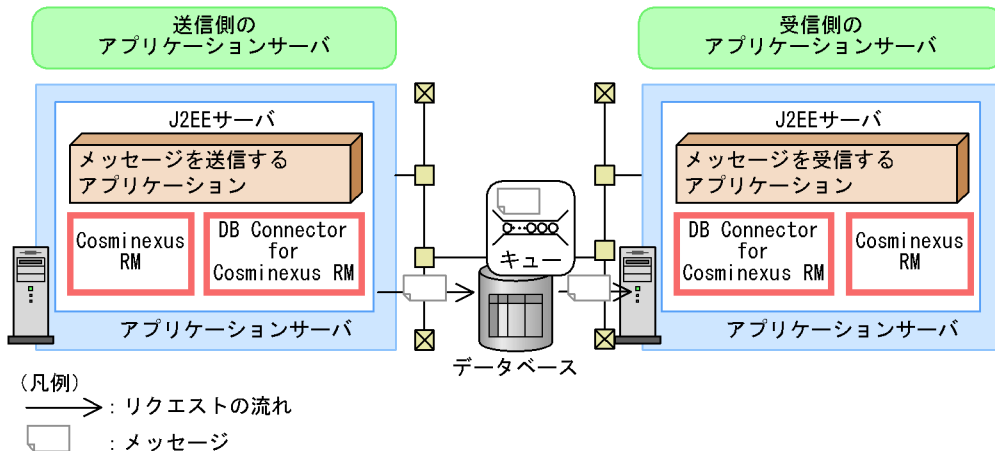
4.5 メッセージを利用した非同期通信アプリケーションの実行環境を構築・運用する

アプリケーション間での非同期通信には、メッセージを使用します。非同期通信で使用するメッセージとは、J2EE の仕様として規定された JMS インタフェースによって実装されたデータです。送信元のアプリケーションからのリクエストがメッセージの形式で送信されると、メッセージは一度キューに登録されます。そのあとで、受信側のアプリケーションによってキューからメッセージが取り出され、処理が実行されます。これによって、送信側と受信側のアプリケーションを同時に動作させる必要のない非同期通信を実現できます。

また、Cosminexus では、キューをデータベース上に構築します。データベースのデータとして登録することによって、メッセージを送信したときに受信側が受信できなかった場合に、正しく受信されるまでキューにメッセージを保持できます。これによって、確実なメッセージ配信を実現できます。なお、アプリケーションサーバでは、Cosminexus が提供する Cosminexus RM および DB Connector for Cosminexus RM という二つのリソースアダプタを利用して、データベース上のキューの実現、およびアプリケーションからデータベース上のキューへのアクセスなどを実現します。

メッセージを利用した非同期通信アプリケーションの実行環境の例を次の図に示します。

図 4-14 メッセージを利用した非同期通信アプリケーションの実行環境の例



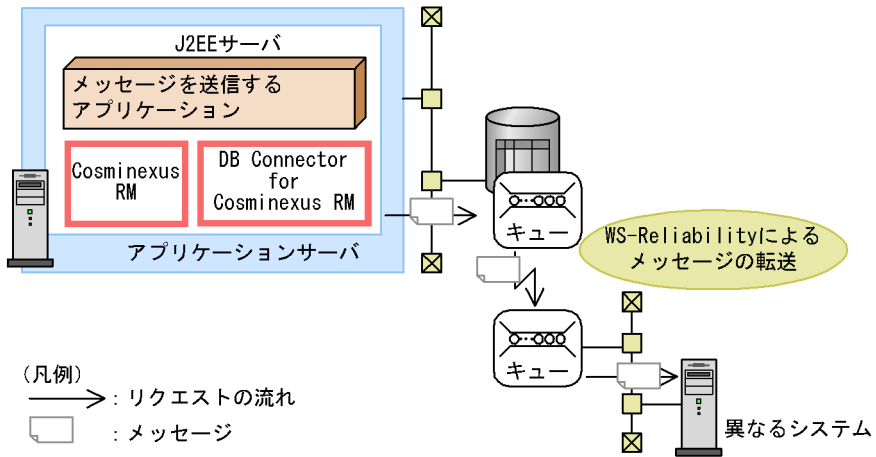
送信側のアプリケーションサーバ上で動作するメッセージを送信するアプリケーションから、Cosminexus RM および DB Connector for Cosminexus RM を経由してメッセージが送信され、キューに登録されます。メッセージがキューに登録されると、受信側のアプリケーションサーバ上の Cosminexus RM および DB Connector for Cosminexus RM によってメッセージが取り出され、メッセージを受信するアプリケーションに渡されます。受信側のアプリケーションサーバでは、メッセージの到着を契機として、アプリケーションが実行されます。なお、同じマシン内のアプリケーション間で非同期通信

を実現したい場合は、送信側のアプリケーションサーバと受信側のアプリケーションサーバを同じマシンにすることもできます。

また、キューに登録されたメッセージを異なるシステムのキューに転送する場合は、WS-Reliability に基づいた転送が実行されます。WS-Reliability は、SOAP プロトコルをベースに、標準化団体 OASIS で標準化された、インターネット向けのオープンな非同期通信プロトコルです。異なるシステム間のキューでメッセージが転送される場合、メッセージは SOAP メッセージの形式で転送されます。

メッセージの転送の概要を、次の図に示します。

図 4-15 メッセージの転送の概要



ポイント

メッセージを利用した非同期通信アプリケーションの実行環境の送信側と受信側に使用できる製品を次の表に示します。

表 4-3 メッセージを利用した非同期通信アプリケーションの送信側と受信側に使用できる製品

構築する実行環境	使用できる製品
送信側	Application Server Standard Application Server Enterprise Service Platform Developer Professional Service Architect
受信側	Application Server Standard Application Server Enterprise Service Platform Developer Professional Service Architect (WS-Reliability に対応した異なるシステム)

4. J2EE アプリケーションを実行する

注 開発環境でのテスト用実行環境として使用できます。

メッセージを利用した非同期通信アプリケーションの実行環境については、マニュアル「Cosminexus Reliable Messaging」を参照してください。

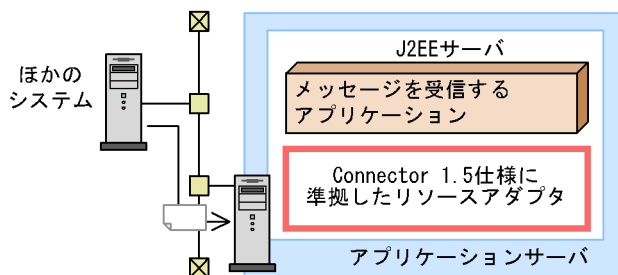
参考

この節の説明は、Cosminexus の構成ソフトウェアである Cosminexus Reliable Messaging を使用した場合の非同期通信の説明です。

このほか、Connector 1.5 仕様に準拠した任意のリソースアダプタを使用して、外部のリソースからのメッセージ送信を受け付け、アプリケーションサーバ上の処理を非同期で実行することもできます。

概要を次の図に示します。

図 4-16 Connector 1.5 仕様に準拠したリソースアダプタを使用する場合の非同期通信



(凡例)

→ : リクエストの流れ

□ : メッセージ

Connector 1.5 仕様に準拠したリソースアダプタを使用したメッセージの非同期通信については、マニュアル「Cosminexus 機能解説」を参照してください。

4.6 導入から運用までの流れ

J2EE アプリケーションを実行する環境の導入から構築・運用までの流れについて説明します。

3章で説明した流れに従って、次の工程を実施します。

システムの仕様（機能）検討

システム設計 / 運用設計

システムの構築

システムの日常運用 / 保守

また、必要に応じて運用の見直しを実施します。

それぞれの作業の概要を説明します。なお、構築・運用する環境によって固有の作業がある場合は、そのつど説明します。

(1) システムの仕様（機能）検討

システムの仕様（機能）検討では、実現したいシステムの要件に応じて、J2EE アプリケーションの実行環境で実現できる機能を確認して、どの機能を使用するかを選択・検討します。

J2EE アプリケーションの実行環境で使用できる機能の詳細については、マニュアル「Cosminexus 機能解説」を参照してください。

また、構築する実行環境に応じて、次のマニュアルも参照してください。

Web サービスに対応したアプリケーション実行環境を構築・運用する場合

マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照して、Cosminexus での SOAP アプリケーション開発の概要、および SOAP アプリケーションの形態について確認し、実現する機能やシステム構成を検討してください

メッセージを利用した非同期通信アプリケーションの実行環境を構築・運用する場合
(Cosminexus Reliable Messaging を使用するとき)

マニュアル「Cosminexus Reliable Messaging」を参照して、メッセージを利用した非同期通信アプリケーションの実行環境で実現する機能やシステム構成を検討してください。

(2) システム設計 / 運用設計

システム設計は、次の考え方で進めます。

1. 使用する機能に応じてシステム構成を検討する。
2. システムの目的に応じたセキュリティの確保の方法を検討する。

4. J2EE アプリケーションを実行する

3. システムの目的に応じたチューニング項目を検討する。

システム設計の考え方の詳細については、マニュアル「Cosminexus システム設計ガイド」を参照してください。

また、構築する実行環境に応じて、次のマニュアルも参照してください。

Web サービスに対応したアプリケーション実行環境を構築・運用する場合

マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照して、SOAP アプリケーションをどのような形態で開発するかを検討し、それに従って、システム設計/運用設計を実施してください。クライアントとサーバのそれぞれについて、運用方法と構築方法を検討してください。

メッセージを利用した非同期通信アプリケーションの実行環境を構築・運用する場合 (Cosminexus Reliable Messaging を使用するとき)

マニュアル「Cosminexus Reliable Messaging」を参照して、実行したい業務処理に応じた送信側、受信側のアプリケーションの形態を検討します。それに従って、システム設計/運用設計を実施してください。クライアントとサーバのそれぞれについて、運用方法と構築方法を検討します。なお、システム構成の検討については、マニュアル「Cosminexus システム設計ガイド」のサーバ間で非同期通信をする場合の構成を検討するための説明を参照してください。

(3) システムの構築

システムの構築は、Smart Composer 機能またはセットアップウィザードを使用して実施します。

システムを構築するときに参照するマニュアルを、次の表に示します。

表 4-4 システムを構築するときに参照するマニュアル

マニュアル	目的	分類
Cosminexus 簡易構築・運用ガイド	Smart Composer 機能またはセットアップウィザードによってシステムを構築する手順について知りたい場合に参照してください。インストール、システム構成に応じたシステム構築方法、J2EE アプリケーションの設定および動作確認といった、一とおりの作業手順について説明しています。 また、テスト環境でチューニングした定義を基に本番環境を構築する場合も、このマニュアルを参照してください。	

マニュアル	目的	分類
Cosminexus システム構築ガイド	マニュアル「Cosminexus 簡易構築・運用ガイド」で説明している手順以外の、特定の機能を使用する場合のシステム構築方法を知りたい場合に参照してください。マニュアル「Cosminexus 機能解説」で説明している機能ごとに必要な設定について説明しています。また、uCosminexus Client, uCosminexus Web Redirector および uCosminexus Operator といった、オプション製品の環境を構築する方法についても、このマニュアルで説明しています。これらのオプション製品を使用するシステムを構築する場合は、このマニュアルを参照してください。	
Hitachi Web Server	Hitachi Web Server のコマンドや定義ファイルを使用して Web サーバを構築したい場合に参照してください。	
TPBroker ユーザーズガイド	Cosminexus TPBroker が使用するポート番号について知りたい場合に参照してください。	

(凡例) : 必ず参照するマニュアル : 必要に応じて参照するマニュアル

なお、これらのマニュアルの中でほかのマニュアルを参照している場合は、必要に応じて参照先のマニュアルも確認してください。

また、構築する実行環境に応じて、次のマニュアルも参照してください。

メッセージを利用した非同期通信アプリケーションの実行環境を構築する場合
(Cosminexus Reliable Messaging を使用するとき)

マニュアル「Cosminexus Reliable Messaging」を参照して、メッセージを利用した非同期通信を実現するために固有な環境構築を実施してください。

(4) システムの日常運用 / 保守

システムの日常運用のうち、システムの起動や停止、および起動状態の確認といった、基本的な運用業務については、Smart Composer 機能を使用して実施します。また、稼働情報・トレースファイルを使用したシステム監視などのシステムの日常的な運用業務、J2EE アプリケーションの入れ替えやトラブルシュートなどの保守的な業務は、Smart Composer 機能以外の運用管理コマンドや、出力されたファイルなどを使用して実施します。

システムを日常運用または保守するときに参照するマニュアルを、次の表に示します。

表 4-5 システムを日常運用または保守するときに参照するマニュアル

マニュアル	目的	分類
Cosminexus 簡易構築・運用ガイド	Smart Composer 機能によってシステムを起動および停止する手順について知りたい場合に参照してください。また、システム内のサービスユニットの稼働状態について確認する場合も、このマニュアルを参照してください。	

4. J2EE アプリケーションを実行する

マニュアル	目的	分類
Cosminexus システム運用ガイド	稼働情報や性能解析トレースの調査方法、J2EE アプリケーションの保守的な運用方法などについて知りたい場合に参照してください。また、トラブルが発生した場合も、このマニュアルを参照して対処してください。	
Hitachi Web Server	次の場合に参照してください。 <ul style="list-style-type: none"> • Hitachi Web Server のコマンドを使用して Web サーバを運用する場合 • Hitachi Web Server のログの詳細を確認する場合 	
TPBroker ユーザーズガイド	Cosminexus TPBroker が使用するログについて知りたい場合に参照してください。	

(凡例) : 必ず参照するマニュアル : 必要に応じて参照するマニュアル

なお、これらのマニュアルの中でほかのマニュアルを参照している場合は、必要に応じて参照先のマニュアルも確認してください。

また、運用する実行環境に応じて、次のマニュアルも参照してください。

メッセージを利用した非同期通信アプリケーションの実行環境を運用する場合
(Cosminexus Reliable Messaging を使用するとき)

マニュアル「Cosminexus Reliable Messaging」を参照して、メッセージを利用した非同期通信を実現するために固有な運用を実施してください。

5

バッチアプリケーションを実行する

この章では、バッチアプリケーションを実行するための環境の概要について説明します。バッチアプリケーションの実行環境は、Application Server Enterprise を使用して構築・運用します。

なお、開発環境用の製品である Developer Professional で構築したテスト用実行環境でも、この章で説明する機能を使用できます。

5.1 バッチアプリケーションの実行環境の特長

5.2 バッチアプリケーションの実行環境を構築・運用する

5.3 導入から運用までの流れ

5.1 バッチアプリケーションの実行環境の特長

この節では、バッチアプリケーションの実行環境の特長について説明します。

5.1.1 オープン環境でのバッチジョブの実行

バッチジョブをオープン環境である Java の実行環境で実行します。

バッチアプリケーションの実行環境は、Java プログラムとして開発されたバッチアプリケーションを実行する環境です。この環境によって、バッチジョブのオープン環境への移行を図れます。

現在、基幹業務の処理を実行するバッチジョブの多くは、メインフレーム上で実行されています。Cosminexus では、バッチジョブの実行環境を、オープン環境である Java の実行環境として構築します。これによって、Java の特長である柔軟性や運用容易性を兼ね備えたバッチジョブの実行環境を構築・運用できます。

バッチアプリケーションは、常駐型の JavaVM プロセスであるバッチサーバで実行します。これによって、バッチジョブを実行するたびに JavaVM を起動するコストを抑えられます。

また、次に示す関連プログラムと連携することで、メインフレームで実現していた内容に類似した処理の制御や自動実行が実現できます。

JP1/AJS2

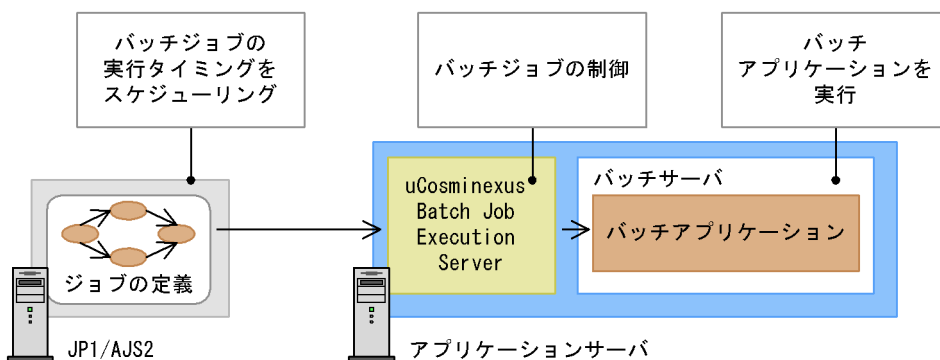
バッチアプリケーションを実行するタイミングをスケジューリングできます。これによって、バッチアプリケーションを使用する業務を自動化できます。

uCosminexus Batch Job Execution Server

メインフレームで実現していたジョブ制御言語によるジョブ管理のイメージで、バッチアプリケーションを実行できます。AIX の場合に使用できます。

バッチアプリケーションの実行環境でのバッチジョブ実行の概要を次の図に示します。

図 5-1 バッチアプリケーションの実行環境でのバッチジョブ実行の概要



この構成の場合は、バッチジョブを実行するタイミングを JP1/AJS2 で制御できます。また、uCosminexus Batch Job Execution Server を使用して定義した内容を基に、バッチジョブを実行できます。

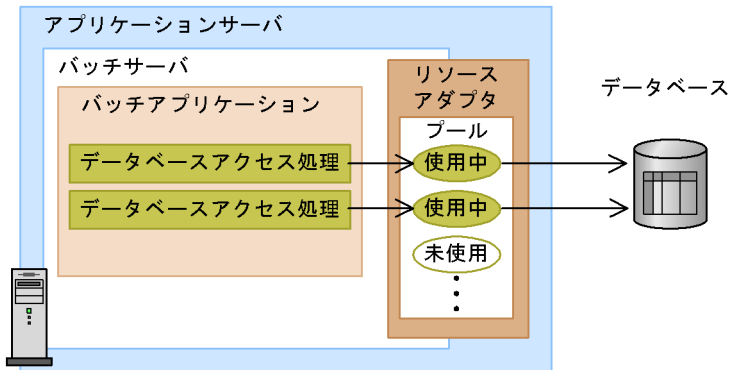
5.1.2 コネクションプール/ステートメントプールを使用したデータベースアクセスの高速化

データベースへの接続で使用するコネクションやステートメントの生成は、時間の掛かる処理です。コネクションプールおよびステートメントプールは、生成したコネクションやステートメントをプールしておくことで、処理性能の向上を図る機能です。

バッチアプリケーションの実行環境では、コネクションプールやステートメントプールなどの機能を使用できます。これによって、バッチアプリケーションからデータベースに高速にアクセスできます。なお、コネクションプールおよびステートメントプールの機能は、リソースアダプタを使用して実現します。

コネクションプールまたはステートメントプールを使用したデータベースアクセスの概要を次の図に示します。

図 5-2 コネクションプールまたはステートメントプールを使用したデータベースアクセスの概要



(凡例)

● および ○ : プールしていたコネクションまたはステートメント

バッチアプリケーションからデータベースにアクセスするときには、プールしていたコネクションまたはステートメントを使用します。

5.1.3 リソース排他状態でのフルガーベージコレクション実行を抑止

JavaVM では、空きメモリが一定のサイズ以下になると、フルガーベージコレクションが実行されます。フルガーベージコレクションが実行されると、その JavaVM 上での処理はすべて中断されます。バッチサーバ上のアプリケーションがリソースを排他状態で

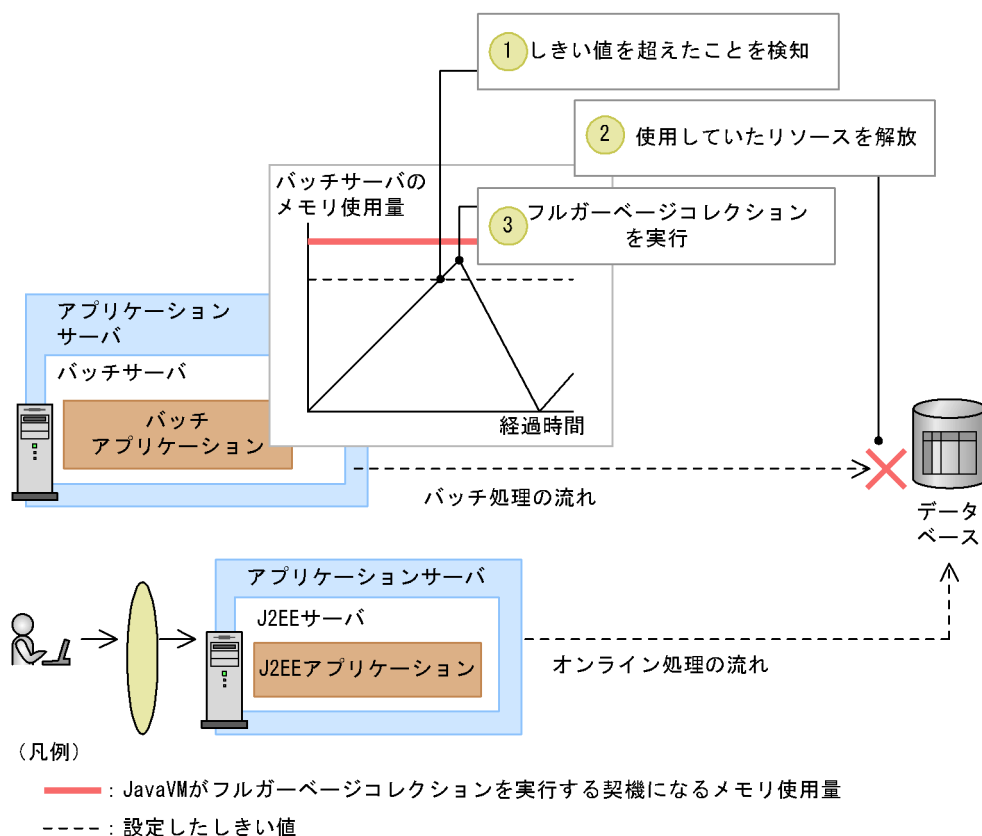
5. バッチアプリケーションを実行する

使用している時にフルガーベージコレクションが実行されると、排他したままで処理が中断されます。この場合、同じリソースを使用したいオンライン処理の実行も中断されてしまいます。

バッチアプリケーションの実行環境では、バッチサーバ上でフルガーベージコレクションを実行するタイミングを制御できます。この制御では、リソースが排他されていないタイミングを見計らってフルガーベージコレクションを実行して、不要なオブジェクトを解放します。これによって、意図しないタイミングでフルガーベージコレクションが実行されることを抑止できます。

バッチサーバでのフルガーベージコレクション制御の概要を次の図に示します。

図 5-3 バッチサーバでのフルガーベージコレクション制御の概要



1. でフルガーベージコレクション制御によって、メモリ使用量がしきい値を超えたことが検知されます。バッチサーバでは、使用していたリソースを解放したことを 2. で確認してから、3. のフルガーベージコレクションを実行します。

5.1.4 そのほかの特長

5.1.3 までに示した以外に、バッチアプリケーションの実行環境には次の特長があります。これらは、J2EE アプリケーションの実行環境と同じです。

可用性と耐障害性の向上

「4.1.3 可用性と耐障害性の向上」を参照してください。ただし、セッション情報の引き継ぎは該当しません。

システム導入および拡張の容易化

「4.1.4 システム導入および拡張の容易化」を参照してください。

システム監査によるシステムのセキュリティ確保

「4.1.5 システム監査によるシステムのセキュリティ確保」を参照してください。

業務効率を向上させる運用管理の実現

「4.1.6 業務効率を向上させる運用管理の実現」を参照してください。ただし、統合ユーザ管理、メンテナンス時のアプリケーション入れ替えの効率化は該当しません。

5.2 バッチアプリケーションの実行環境を構築・運用する

この節では、バッチアプリケーションの実行環境の構築・運用について説明します。

5.2.1 システム構成の種類

ここでは、バッチアプリケーションの実行環境のシステム構成について説明します。

バッチアプリケーションの実行環境は、バッチサーバを配置したアプリケーションサーバを中心に構築します。また、業務の用途に合わせて、バッチジョブを自動実行するための関連プログラムを配置します。

ここでは、バッチアプリケーションの実行環境として、次の2種類のシステム構成について説明します。

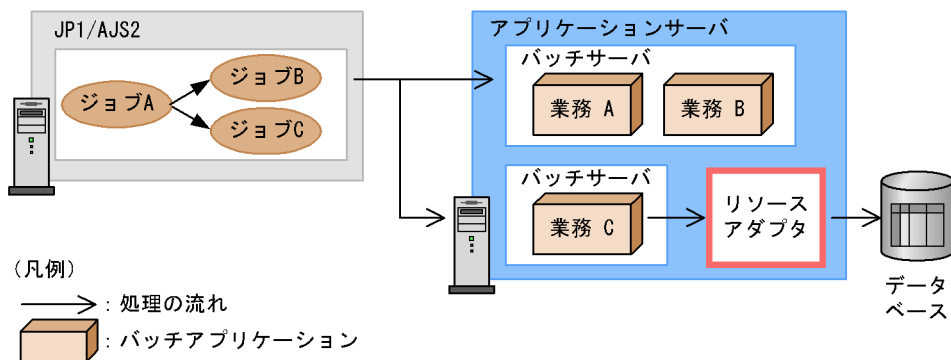
バッチ処理だけを実行するシステム

バッチ処理とオンライン処理を同時に実行するシステム

(1) バッチ処理だけを実行するシステム

バッチ処理だけを実行するシステムの構成例を次の図に示します。

図 5-4 バッチ処理だけを実行するシステムの構成例



この構成例では、1台のアプリケーションサーバ内に二つのバッチサーバを構築しています。それぞれのバッチサーバに対して、それぞれ異なるバッチアプリケーションを配置します。なお、一つのバッチサーバの中で複数のバッチアプリケーションを同時に実行することはできません。この図の場合、業務 A と業務 B は同時に実行されることはありません。業務 C は、リソースに接続するバッチアプリケーションです。

また、バッチサーバから、ほかの J2EE サーバ上で動作する J2EE アプリケーション (Enterprise Bean) を呼び出すこともできます。その場合、Enterprise Bean を実行するアプリケーションサーバは、フロントエンド/バックエンドシステムのバックエンド

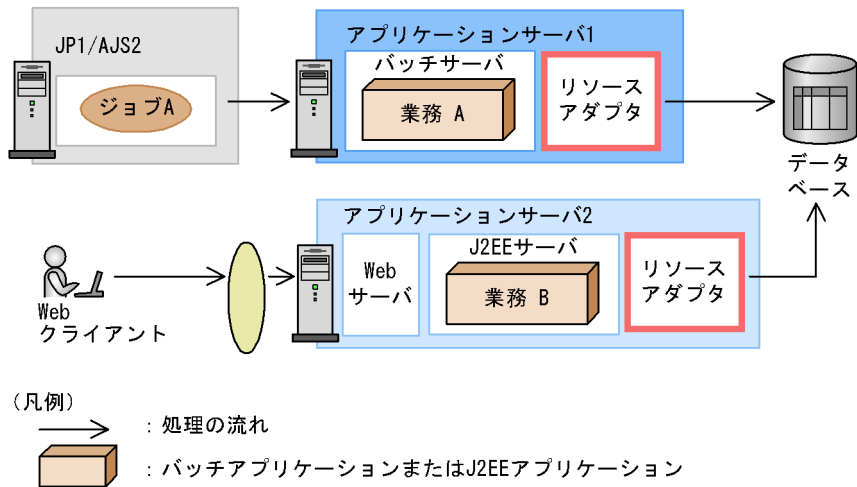
に該当します。バックエンドシステムについては、「4.3.1(2) フロントエンド/バックエンドシステム」を参照してください。

JP1 ジョブ運用管理サーバは、バッチジョブを自動実行する場合に必要な関連プログラムをインストールしたマシンです。必要に応じて配置します。

(2) バッチ処理とオンライン処理を同時に実行するシステム

一つのシステム内でバッチ処理とオンライン処理を同時に実行するシステムの構成例を次の図に示します。

図 5-5 バッチ処理とオンライン処理を同時に実行するシステムの構成例



この構成では、バッチ処理を実行するアプリケーションサーバ1と、オンライン処理を実行するアプリケーションサーバ2を構築します。また、バッチサーバ上のバッチアプリケーションと、J2EEサーバ上のJ2EEアプリケーションで同じリソースにアクセスします。

JP1 ジョブ運用管理サーバは、バッチジョブを自動実行する場合に必要な関連プログラムをインストールしたマシンです。必要に応じて配置します。

5.2.2 バッチアプリケーションの実行環境の構築・運用

バッチアプリケーションの実行環境の構築・運用には、Smart Composer 機能を使用できます。Smart Composer 機能を使用すると、バッチアプリケーションの実行環境を、少ない手順で一括して構築・運用できます。

Smart Composer 機能では、一つの業務処理を実行するサーバの集合を、サービスユニットとして扱います。また、バッチサーバをJ2EEサーバとして扱います。

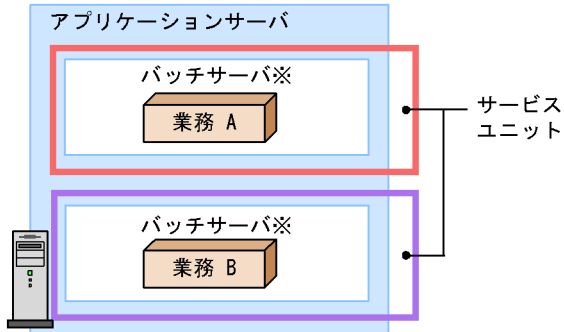
Smart Composer 機能でバッチアプリケーションの実行環境を構築する場合は、サービ

5. バッチアプリケーションを実行する

スユニット内に J2EE サーバだけを構築するシステム構成パターンを選択して構築します。また、一部のパラメタに、バッチサーバ特有の値を設定します。

Smart Composer 機能によって構築するバッチアプリケーションの実行環境の例を、次の図に示します。

図 5-6 Smart Composer 機能によって構築するバッチアプリケーションの実行環境の例



注※ Smart Composer機能ではJ2EEサーバとして扱います。

5.3 導入から運用までの流れ

バッチアプリケーションを実行する環境の導入から構築・運用までの流れについて説明します。

3章で説明した流れに従って、次の工程を実施します。

システムの仕様（機能）検討

システム設計 / 運用設計

システムの構築

システムの日常運用 / 保守

また、必要に応じて運用の見直しを実施します。

それぞれの作業の概要を説明します。なお、構築・運用する環境によって固有の作業がある場合は、そのつど説明します。

(1) システムの仕様（機能）検討

システムの仕様（機能）検討では、実現したいシステムの要件に応じて、バッチアプリケーションの実行環境で実現できる機能を確認して、どの機能を使用するかを選択・検討します。

バッチアプリケーションの実行環境で使用できる機能の詳細については、マニュアル「Cosminexus 機能解説」を参照してください。

(2) システム設計 / 運用設計

システム設計は、次の考え方で進めます。

1. 使用する機能に応じてシステム構成を検討する。
2. システムの目的に応じたチューニング項目を検討する。

システム設計の考え方の詳細については、マニュアル「Cosminexus システム設計ガイド」を参照してください。

(3) システムの構築

システムの構築は、Smart Composer 機能を使用して実施します。

システムを構築するときに参照するマニュアルを、次の表に示します。

5. バッチアプリケーションを実行する

表 5-1 システムを構築するときに参照するマニュアル

マニュアル	目的	分類
Cosminexus 簡易構築・運用ガイド	Smart Composer 機能によってシステムを構築する手順について知りたい場合に参照してください。インストール、システム構成に応じたシステム構築方法および動作確認といった、一とおりの作業手順について説明しています。 また、テスト環境でチューニングした定義を基に本番環境を構築する場合も、このマニュアルを参照してください。	
Cosminexus システム構築ガイド	マニュアル「Cosminexus 簡易構築・運用ガイド」で説明している手順以外の、特定の機能を使用する場合のシステム構築方法を知りたい場合に参照してください。マニュアル「Cosminexus 機能解説」で説明している機能ごとに必要な設定について説明しています。	
TPBroker ユーザーズガイド	Cosminexus TPBroker が使用するポート番号について知りたい場合に参照してください。	

(凡例) : 必ず参照するマニュアル : 必要に応じて参照するマニュアル

なお、これらのマニュアルの中でほかのマニュアルを参照している場合は、必要に応じて参照先のマニュアルも確認してください。

(4) システムの日常運用 / 保守

システムの日常運用のうち、システムの起動や停止、および起動状態の確認といった、基本的な運用業務については、Smart Composer 機能を使用して実施します。また、稼働情報・トレースファイルを使用したシステムの監視などのシステムの日常的な運用業務やトラブルシュートなどの保守的な業務は、Smart Composer 機能以外の運用管理コマンドや、出力されたファイルなどを使用して実施します。

システムを日常運用または保守するときに参照するマニュアルを、次の表に示します。

表 5-2 システムを日常運用または保守するときに参照するマニュアル

マニュアル	目的	分類
Cosminexus 簡易構築・運用ガイド	Smart Composer 機能によってシステムを起動および停止する手順について知りたい場合に参照してください。また、システム内のサービスユニットの稼働状態について確認する場合も、このマニュアルを参照してください。	
Cosminexus システム運用ガイド	稼働情報や性能解析トレースの調査方法について知りたい場合に参照してください。また、トラブルが発生した場合も、このマニュアルを参照して対処してください。	
TPBroker ユーザーズガイド	Cosminexus TPBroker が使用するログについて知りたい場合に参照してください。	

(凡例) : 必ず参照するマニュアル : 必要に応じて参照するマニュアル

なお、これらのマニュアルの中でほかのマニュアルを参照している場合は、必要に応じて参照先のマニュアルも確認してください。

6

アプリケーションを開発する

Cosminexus では、実行環境で動作するアプリケーションの開発を支援するための機能を、開発環境で提供しています。アプリケーションの開発環境は、Developer Standard、Developer Professional または Service Architect を使用して構築します。この章では、Cosminexus の開発環境が提供する開発支援機能を使用したアプリケーションの開発について説明します。

6.1 Cosminexus でのアプリケーション開発の特長

6.2 開発するアプリケーションの種類

6.3 J2EE アプリケーションを開発する

6.4 Web サービスに対応したアプリケーションを開発する

6.5 メッセージを使用した非同期通信用のアプリケーションを開発する

6.6 アプリケーション開発の流れ

6.1 Cosminexus でのアプリケーション開発の特長

Cosminexus では、実行環境上で動作するアプリケーションを、わかりやすい操作で効率良く開発できるようにするための開発環境を構築できます。

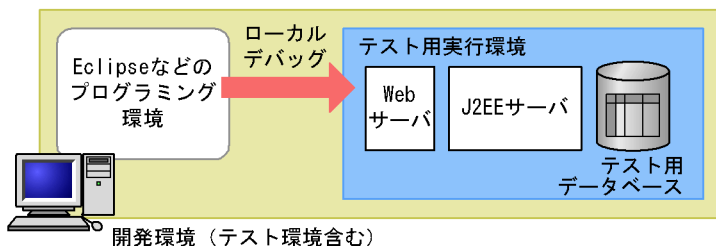
IDE として Eclipse を使用してアプリケーションを開発する場合に、Cosminexus で提供するプラグインを組み込んで使用することで、アプリケーション開発時の一連の作業をすべて Eclipse、または Eclipse のプラグインで実現できるようになります。

また、Cosminexus の開発環境では、開発したアプリケーションのテスト環境として、実行環境と同じ機能を持つ環境を構築できます。このため、アプリケーションの設計、プログラミングからテスト環境でのテスト、デバッグまでの一連の流れを、すべて Cosminexus の開発環境上で実施できます。

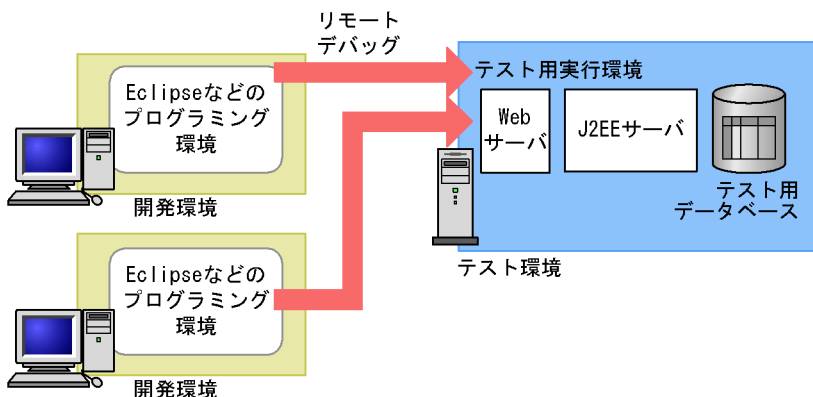
開発環境は、アプリケーションの開発方法やデバッグの実行方法に応じて、次の図に示す構成で構築できます。

図 6-1 Cosminexus の開発環境の構成

- 開発環境とテスト環境を一つのマシンに構築する構成



- 開発環境とテスト環境を別のマシンに構築する構成



開発環境とテスト環境を一つのマシンに構築する構成

アプリケーションを一人の開発者が1台のマシンで開発する場合などに適しています。ローカルデバッグを実行できます。

開発環境とテスト環境を別のマシンに構築する構成

アプリケーションを複数の開発者のチームで開発する場合などに適しています。リモートデバッグを実行できます。なお、実行環境がWindows以外のOSの場合のテストを実行するときには、この構成にします。

それぞれの場合にインストールする製品を次の表に示します。

表 6-1 開発環境の構成ごとにインストールする製品

構成	開発環境	テスト環境
開発環境とテスト環境を一つのマシンに構築する構成	<ul style="list-style-type: none"> • Developer Standard • Developer Professional • Service Architect 	
開発環境とテスト環境を別のマシンに構築する構成	<ul style="list-style-type: none"> • Developer Professional • Service Architect 	<ul style="list-style-type: none"> • Application Server Standard • Application Server Enterprise • Service Platform

ポイント

Developer Standard は、一つの J2EE サーバ内で動作するアプリケーションを開発するための製品です。提供されるテスト環境も、1台のアプリケーションサーバ内での動作を前提としています。このため、テスト環境をほかのマシンに構築する構成には対応していません。

このほか、Developer Standard と Developer Professional では、使用できる MyEclipse の機能も一部異なります。

Developer Standard と Developer Professional の機能差については、マニュアル「Cosminexus アプリケーション開発ガイド」を参照してください。

6.2 開発するアプリケーションの種類

Cosminexus では、J2EE アプリケーション、Web サービスに対応したアプリケーション、およびメッセージを使用した非同期通信のアプリケーションを開発できます。それぞれのアプリケーション開発の概要を説明します。

J2EE アプリケーションの開発

JSP、サーブレット、Enterprise Bean を作成し、J2EE アプリケーションを開発できます。Cosminexus では、J2EE アプリケーションの開発に MyEclipse を利用できます。MyEclipse を使用すると、JSP、サーブレット、Enterprise Bean の作成からテスト・デバッグまでシームレスなアプリケーション開発が実現します。

J2EE アプリケーションの開発については、「6.3 J2EE アプリケーションを開発する」で説明します。

Web サービスに対応したアプリケーションの開発

Web サービスに対応したアプリケーション（SOAP アプリケーション）を開発できます。Cosminexus では、Web サービスに対応したアプリケーションの、開発および実行に必要なファイルやソースコード（スタブやスケルトンなど）を自動生成できます。SOAP アプリケーションの開発については、「6.4 Web サービスに対応したアプリケーションを開発する」で説明します。

メッセージを使用した非同期通信のアプリケーションの開発

メッセージを使用した非同期通信のアプリケーションを開発できます。

Cosminexus が提供する JMS インタフェースを利用してアプリケーションを開発できます。

非同期通信のアプリケーションの開発については、「6.5 メッセージを使用した非同期通信のアプリケーションを開発する」で説明します。

6.3 J2EE アプリケーションを開発する

この節では、Cosminexus の機能を使用した J2EE アプリケーションの開発の概要について説明します。

6.3.1 Developer が提供する MyEclipse およびプラグインを使用する

Cosminexus の開発環境では、Developer が提供する MyEclipse および Server Plug-in を使用して、アプリケーションを開発できます。それぞれについて説明します。

MyEclipse

Eclipse でアプリケーションを開発するとき使用するプラグインがまとめられた、Java アプリケーションの開発環境です。Java アプリケーションの上流設計から、アプリケーションの開発、テスト、デプロイまで、シームレスに実行できます。

Server Plug-in

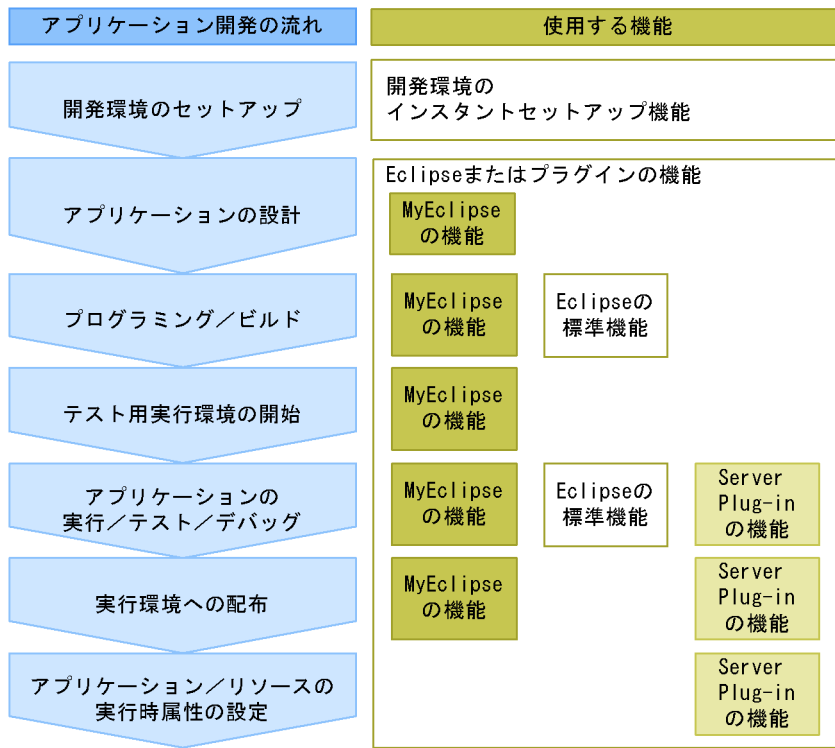
テスト環境上でのアプリケーションやリソースの操作、およびテスト環境の運用を支援する Eclipse のプラグインです。

MyEclipse、および Server Plug-in は Eclipse に組み込んで使用します。これによって、統合された環境で、効率良くアプリケーションを開発できます。

アプリケーション開発の流れと Developer で使用する機能の関係について、次の図に示します。

6. アプリケーションを開発する

図 6-2 アプリケーション開発の流れと使用する機能の関係



アプリケーション開発で実施する作業について説明します。

開発環境のセットアップ

開発環境に必要な製品をインストールして、開発環境をセットアップします。開発環境のセットアップには、Developer が提供する開発環境のインスタントセットアップ機能を使用できます。開発環境のインスタントセットアップ機能は、アプリケーションの開発およびデバッグを実行する環境を構築する機能です。アプリケーションの開発・デバッグをするために必要なサーバプロセスとデータベースの環境構築をウィザードプログラムのダイアログから実行できます。

アプリケーションの設計

MyEclipse では UML を使用した設計をサポートしています。UML を使用した設計を実施する場合は、MyEclipse の UML パースペクティブを利用できます。

プログラミング/ビルド

MyEclipse を使用してプロジェクトを作成し、JSP、サーブレット、および Enterprise Bean を作成します。なお、データベースなどのリソース接続に必要なリソースアダプタも、MyEclipse のプロジェクトに含めることができます。MyEclipse の場合、アプリケーションのビルドは、アプリケーションのデプロイ時にあわせて実行されます。

テスト用実行環境の開始

MyEclipse を使用して、テスト用 J2EE サーバを開始します。また、テスト用のデータベースを開始します。なお、開発環境のインスタントセットアップ機能を使用して構築したシステムの場合は、スタートメニューからデータベースを開始または停止できます。それ以外の場合は、Developer で提供するコマンドを使用して、データベースを開始します。

アプリケーションの実行 / テスト / デバッグ

MyEclipse を使用して、アプリケーションを J2EE サーバ上にデプロイします。アプリケーションのテストでデータベースなどに接続する場合は、Server Plug-in を使用して、J2EE リソースのプロパティを設定します。

また、アプリケーションのデバッグには、Eclipse のデバッグ機能を使用します。

実行環境への配布

MyEclipse を使用して、アプリケーションを EAR 形式でエクスポートします。エクスポートした EAR ファイルは、Server Plug-in を使用して、実行環境にインポートします。

アプリケーション / リソースの実行時属性の設定

Server Plug-in を使用して、実行環境にインポートしたアプリケーションに実行時属性を設定します。なお、開発環境のインスタントセットアップ機能を使用して構築したシステムの場合は、データベースへの接続に必要な設定をしたリソースアダプタがデプロイされています。設定内容を変更したい場合や、ほかのリソースアダプタを使用したい場合には、必要に応じて Server Plug-in を使用して操作してください。

6.3.2 そのほかに使用できる機能

Cosminexus では、次のアプリケーションを開発するための機能も提供しています。

JAXP で規定された API を利用したアプリケーションの開発

J2EE 環境で動作する CORBA クライアントアプリケーションの開発

これらの機能の詳細については、次のマニュアルを参照してください。

- Cosminexus XML Processor ユーザーズガイド
- Borland(R) Enterprise Server VisiBroker(R) デベロッパーズガイド
- Borland(R) Enterprise Server VisiBroker(R) プログラマーズリファレンス

注 このマニュアルに記載されている機能のうち、Cosminexus で使用できる機能については、リリースノートを参照してください。

6.4 Web サービスに対応したアプリケーションを開発する

この節では、Web サービスに対応したアプリケーションである、SOAP アプリケーションの開発時に使用できる機能の概要について説明します。

6.4.1 SOAP アプリケーションの開発支援機能を使用する

SOAP アプリケーションは、Web の標準技術を利用してネットワーク上でサービスを公開、実行できるアプリケーションです。

Cosminexus で開発できる SOAP アプリケーションの種類は、大きく分けて、次の 3 種類があります。

RPC を利用した SOAP アプリケーション

RPC によって SOAP メッセージをやり取りするアプリケーションです。RPC を利用した SOAP アプリケーションでは WSDL が必要になります。Cosminexus では、WSDL 生成を支援する機能を使用できます。また、生成した WSDL から、実行環境での SOAP メッセージの送受信に必要なスタブおよびスケルトンも自動生成できます。

メッセージングを利用した SOAP アプリケーション

スタブおよびスケルトンを使用しないで、XML データである SOAP メッセージを直接やり取りするアプリケーションです。XML データを直接やり取りするため、実行時には、Cosminexus が提供する API を利用して、メッセージ生成処理をします。

EJB を利用した SOAP アプリケーション

既存または新規の EJB を Web サービスとして利用するアプリケーションです。EJB を利用した SOAP アプリケーションでは WSDL が必要になります。

Cosminexus では、WSDL 生成を支援する機能を使用できます。また、生成した WSDL から、実行環境での SOAP メッセージの送受信に必要なスタブも自動生成できます。

SOAP アプリケーションを利用してネットワーク上のサービスを利用する場合、クライアント側とサーバ側に、それぞれ次の実装が必要です。

クライアント側

SOAP サービスを呼び出す処理、または SOAP サービスを呼び出すためのメッセージを生成する処理

サーバ側

SOAP サービス

Cosminexus の開発環境では、SOAP アプリケーションの開発および実行に必要なファ

イルやソースコードを自動生成できます。新規に SOAP アプリケーションを開発する場合には、必要な定義ファイルおよびサーバとクライアントの実装の基になるスタブとスケルトンを自動生成できます。また、既存のアプリケーションを SOAP サービスにする場合は、必要な定義ファイルを自動生成できます。これによって、アプリケーション開発者は業務ロジックの実装に注力できます。

Cosminexus で開発できる SOAP アプリケーションについては、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。

また、SOAP アプリケーションでは、UDDI レジストリにアクセスするための UDDI クライアントライブラリも使用できます。

参考

UDDI レジストリの利用について

ネットワーク上の Web サービスが増えてきた場合、Web サービスを登録・検索するための情報の格納先として、UDDI レジストリを利用できます。Cosminexus で開発した SOAP アプリケーションを Web サービスとして UDDI レジストリに登録することもできます。

UDDI レジストリへの登録・検索方法については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。UDDI レジストリの構築・運用方法については、マニュアル「uCosminexus UDDI Registry システム運用ガイド」を参照してください。

6.4.2 Web サービスセキュリティ技術を使用する

ここでは、SOAP アプリケーションを開発する時に使用できる Web サービスセキュリティ技術について説明します。

ポイント

Web サービスセキュリティ技術を使用したアプリケーションは、Developer Professional または Service Architect を使用している場合に開発できます。また、開発したアプリケーションは、Application Server Enterprise または Service Platform を使用して構築された実行環境で実行できます。

Web サービスセキュリティ技術は、SOAP アプリケーションで送受信する XML 形式のデータを、改ざんされたり第三者から盗聴されたりすることを防ぐための技術です。また、W3C が規定している XML 署名および XML 暗号の仕様である、XML セキュリティの技術と組み合わせたセキュリティ確保も実現できます。

Cosminexus が提供する API を使用して SOAP アプリケーションを実装することで、Web サービスセキュリティ技術を使用したアプリケーションを開発できます。

また、Cosminexus では、次の SOAP アプリケーションも開発できます。

6. アプリケーションを開発する

XML 署名データを利用するアプリケーション

XML 暗号によってデータの暗号化 / 復号化を実施するアプリケーション

XML 署名データを利用することによって、電子データの完全性を保証したり、成り済ましたり否認を防止したりできます。XML 暗号を利用することによって、データの秘匿性を保証したり、鍵合意によって共通鍵を安全に作成したりできます。

Web サービスセキュリティの詳細については、マニュアル「Cosminexus Web サービスセキュリティ 使用の手引」を参照してください。また、XML 署名および XML 暗号については、マニュアル「Cosminexus XML Security - Core ユーザーズガイド」を参照してください。

6.5 メッセージを使用した非同期通信用のアプリケーションを開発する

メッセージを使用した非同期通信を実行するには、送信側のアプリケーションと受信側のアプリケーションが必要です。これらのアプリケーションは、Cosminexus が提供する JMS インタフェースを使用して開発できます。また、扱うメッセージは、JMS インタフェースを使用した JMS メッセージです。JMS メッセージは Message インタフェースをルートインタフェースとするオブジェクトです。

送信側および受信側のアプリケーションは、実現する業務内容に合わせて、サーブレット、JSP または Enterprise Bean として作成します。受信側のアプリケーションを Message-driven Bean として作成することで、受信用のキューにメッセージが登録されたことを契機にメッセージを受け取り、必要な業務処理を実行するアプリケーションを開発できます。

非同期通信用アプリケーションの開発については、マニュアル「Cosminexus Reliable Messaging」を参照してください。

参考

この節の説明は、Cosminexus の構成ソフトウェアである Cosminexus Reliable Messaging を使用した場合の非同期通信の説明です。

このほか、Connector 1.5 仕様に準拠した任意のリソースアダプタを使用して、外部のリソースからのメッセージ送信を受け付けて非同期で実行することもできます。この場合のアプリケーションの開発方法については、使用するリソースアダプタの仕様に従ってください。

6.6 アプリケーション開発の流れ

Cosminexus のアプリケーション開発環境での、開発環境の準備とアプリケーション開発の流れについて説明します。次の図に、アプリケーションの開発の流れと、参照するマニュアルの対応を示します。

図 6-3 アプリケーション開発の流れと参照マニュアルの対応



注 マニュアル名称の「Cosminexus」は省略しています。

注※ アプリケーションの形式によっては不要です。

それぞれの作業の概要を説明します。なお、作成するアプリケーションによって固有の作業がある場合は、そのつど説明します。

(1) 開発環境のセットアップ

開発環境に必要な製品 (Developer Standard , Developer Professional または Service Architect) をインストールして、開発環境およびテスト用実行環境をセットアップします。開発環境およびテスト用実行環境のセットアップには、開発環境のインスタントセットアップ機能を使用できます。この機能によって、次の環境をセットアップできます。

開発に必要な統合開発環境 (Eclipse および MyEclipse)

テスト用の実行環境 (J2EE サーバを含むアプリケーションサーバおよび組み込み

データベース)

ただし、開発環境のインスタントセットアップ機能で構築できる環境は、アプリケーションを開発・デバッグするために必要な最小限の環境です。Web サーバや CTM など、開発環境のインスタントセットアップ機能の対象に含まれない機能を使用したテストを実行したい場合には、Smart Composer 機能を使用してシステムを構築してください。

開発環境のインスタントセットアップ機能については、マニュアル「Cosminexus アプリケーション開発ガイド」を参照してください。Smart Composer 機能については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

(2) アプリケーションの設計

実現したい業務の内容に応じて適切なアプリケーションの形態を検討、設計します。また、フレームワークを使用する場合は、どのフレームワークを使用するかも検討します。Web アプリケーションを開発する場合は、画面のレイアウトや画面遷移についても設計しておきます。

なお、UML を使用してアプリケーションを設計する場合は、MyEclipse の機能を利用することもできます。

(3) プログラミング/コンパイル/デバッグ

設計した内容に応じて、サーブレット、JSP、Enterprise Bean など処理を実装して、コンパイルします。実装やコンパイルには、Eclipse などを使用します。

また、アプリケーションの内容に応じた DD も定義します。コンパイルで問題が発生した場合はデバッグを実施します。

Cosminexus の実行環境で動作するアプリケーションのプログラミングで留意する点については、マニュアル「Cosminexus 機能解説」を参照してください。

また、アプリケーション開発で使用する機能やアプリケーションの種類に応じて、次のマニュアルも参照してください。

J2EE アプリケーションを開発する場合

JSP、サーブレット、および Enterprise Bean を作成します。

- JSP およびサーブレットの作成

MyEclipse を使用すると、ひな型を利用して JSP およびサーブレットを作成できます。また、MyEclipse のビジュアル JSP デザイナーを使用することで、グラフィカルに JSP を編集できます。

- Enterprise Bean の作成

Enterprise Bean の実装には、Eclipse の標準機能を使用できます。

J2EE アプリケーションを開発する場合は、マニュアル「Cosminexus アプリケーション開発ガイド」を参照してください。

6. アプリケーションを開発する

SOAP アプリケーションを開発する場合

サーバ側とクライアント側のアプリケーションの実装が必要です。また、実装の前提として、SOAP アプリケーション固有の作業として、次のような作業が必要です。

- SOAP アプリケーションのインタフェースの作成
- WSDL の作成
- サーバスケルトンの作成（サーバ側のアプリケーション）
- クライアントスタブの作成（クライアント側のアプリケーション）

なお、必要な作業は SOAP アプリケーションの形態によって異なります。

Cosminexus の機能を使用して、サーバスケルトンおよびクライアントスタブを WSDL から自動生成する方法については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。

また、Web サービスセキュリティの技術を使用する場合は、マニュアル

「Cosminexus Web サービスセキュリティ 使用の手引」を参照してください。

XML 署名および XML 暗号を使用する場合は、マニュアル「Cosminexus XML Security - Core ユーザーズガイド」を参照してください。

メッセージを使用した非同期通信のアプリケーションを開発する場合（Cosminexus Reliable Messaging を使用するとき）

JMS インタフェースまたは転送データ相互接続用インタフェースを使用してメッセージを操作するアプリケーションを開発します。アプリケーションの開発方法、および Cosminexus RM が提供するインタフェースについては、マニュアル「Cosminexus Reliable Messaging」を参照してください。

（４）アーカイブファイルの作成

実装したサーブレット、JSP、Enterprise Bean を、WAR または EJB-JAR の形式にアーカイブします。さらに、WAR および EJB-JAR を EAR の形式にアーカイブします。アーカイブを作成するには、Eclipse などの IDE や、jar コマンドを使用します。

一度、J2EE サーバにデプロイしたアプリケーションは、サーバ管理コマンドという CUI を使用すると、EAR 形式でアーカイブして、J2EE サーバから取り出すことができます。Eclipse の操作方法については、マニュアル「Cosminexus アプリケーション開発ガイド」を参照してください。サーバ管理コマンドの操作方法については、マニュアル「Cosminexus アプリケーション設定操作ガイド」を参照してください。

また、Cosminexus では、アーカイブファイルを作成しないで、ディレクトリに展開されたファイル群をアプリケーション（展開ディレクトリ形式のアプリケーション）として J2EE サーバにインポートすることもできます。この形式を利用する場合は、アーカイブファイルの作成は不要です。

ポイント

MyEclipse でアプリケーションを開発する場合、アプリケーション開発中にアプリケーションの形式（アーカイブ形式または展開ディレクトリ形式）を意識する必要はありません。MyEclipse でアプリケーションをデプロイするときに形式を指定すると、アプリケーションは、指定した形式で J2EE サーバにデプロイされます。あらかじめ、アーカイブファイルを作成しておく必要はありません。

(5) テスト用実行環境の開始

アプリケーションのテストを実行するための環境を準備します。J2EE サーバなど、アプリケーションサーバで必要なプロセスを起動します。また、必要に応じて、データベースなど連携するシステムも起動します。

テスト用実行環境で必要なサーバプロセスの起動は、MyEclipse または Server Plug-in で実行できます。MyEclipse を使用したアプリケーション開発の延長でテストを実施する場合は、MyEclipse から起動すると便利です。ただし、MyEclipse から起動できるプロセスは、J2EE サーバだけとなります。Web サーバなどと一緒に J2EE サーバを起動したい場合は、Server Plug-in を使用します。

アプリケーションのデプロイ、実行、テストは、MyEclipse、Server Plug-in またはサーバ管理コマンドで実行できます。また、開発環境のインスタントセットアップ機能で構築したシステムの場合、テストに使用する組み込みデータベースの開始・停止などの操作は、スタートメニューから実行できます。それ以外の場合は、Developer で提供するコマンドを使用して、データベースを開始できます。

MyEclipse については、マニュアル「Cosminexus アプリケーション開発ガイド」を参照してください。Server Plug-in およびサーバ管理コマンドの操作方法については、マニュアル「Cosminexus アプリケーション設定操作ガイド」を参照してください。

(6) 実行 / テスト / デバッグ

テスト用実行環境にアプリケーションをデプロイ、実行して、テストを実施します。テストで問題があった場合はデバッグを実施します。

テスト用実行環境でのアプリケーションの実行、テストについては、MyEclipse、Server Plug-in またはサーバ管理コマンドで実行できます。MyEclipse でのアプリケーションの実行、テストについては、マニュアル「Cosminexus アプリケーション開発ガイド」を参照してください。Server Plug-in およびサーバ管理コマンドの実行方法については、マニュアル「Cosminexus アプリケーション設定操作ガイド」を参照してください。

6. アプリケーションを開発する

ポイント

展開ディレクトリ形式のアプリケーションの場合、一度デプロイしたアプリケーションをデバッグで修正したときには、リロードによってアプリケーションを再読み込みできます。リロードをする場合、デバッグ時のアプリケーションの停止、削除、インポート後の再開などの作業が不要になります。アプリケーションのリロードの詳細については、マニュアル「Cosminexus 機能解説」を参照してください。

(7) アプリケーション/リソースの実行時属性の設定

開発したアプリケーションを実行環境の J2EE サーバにインポートして、必要な実行時属性を設定します。また、アプリケーションで利用するリソースについても必要に応じてインポートして実行時属性を設定します。

アプリケーション/リソースの実行時属性の設定は、Eclipse に組み込んだ Server Plug-in またはサーバ管理コマンドで実行できます。Server Plug-in およびサーバ管理コマンドの操作方法については、マニュアル「Cosminexus アプリケーション設定操作ガイド」を参照してください。

7

サービスの統合を実現する

この章では、Cosminexus で実現できるサービス統合について説明します。

Cosminexus のサービス統合環境は、Cosminexus サービスプラットフォーム (Service Platform および Service Architect) として構築します。

7.1 サービス統合環境の特長

7.2 メッセージフローによってサービスを統合する

7.3 メッセージフローによるサービス統合に必要なコンポーネントを定義 / 開発する

7.1 サービス統合環境の特長

この節では、Cosminexus のサービス統合環境の特長について説明します。

7.1.1 SOA の概念に基づくサービスインテグレーションの実現

ビジネス環境の変化に合わせて、ビジネス戦略やビジネスプロセスのライフサイクルが短くなってきています。この状況に即応するためには、変化に柔軟に対応できる、拡張性の高い業務システムが必要です。

このような業務システムを実現するための概念が、SOA です。

SOA では、ビジネス環境の変化に対して、常に新しいアプリケーションを開発して対応するのではなく、既存のアプリケーションを再利用したり組み合わせを変えたりして、新しいサービスとして利用します。既存のアプリケーションを活用するため、短期間で信頼性の高いサービスの提供が可能です。

Cosminexus では、メッセージフローによって、サービス統合を実現します。

ここでは、それぞれの概要について説明します。また、Cosminexus でのサービス統合の考え方についても説明します。

(1) メッセージフローによるサービス統合

Cosminexus サービスプラットフォームは、SOA の概念でサービスの統合（サービスインテグレーション）を実現するための基盤である、エンタープライズ・サービス・バス（ESB）を構築する製品です。Cosminexus のアプリケーション実行環境の機能を基盤として、信頼性と性能の高い ESB を構築できます。

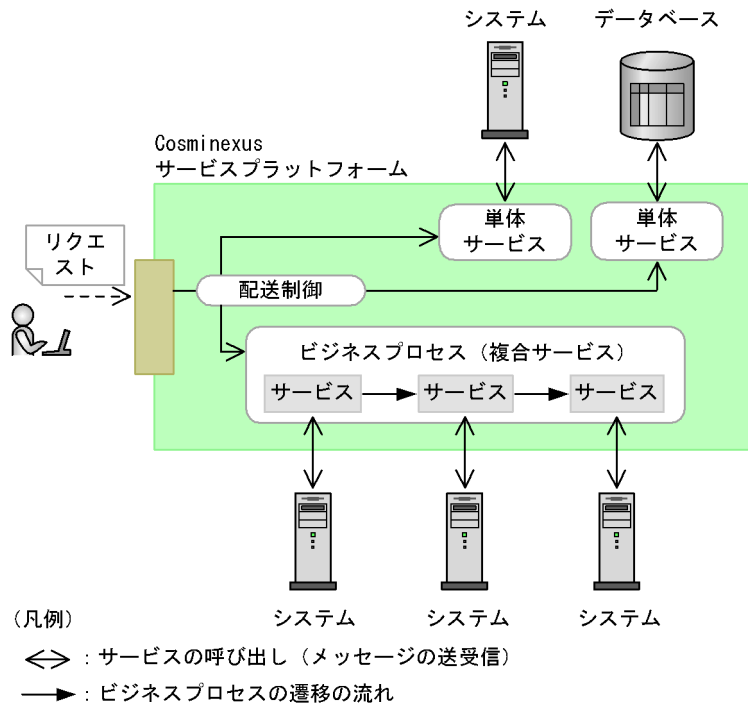
Cosminexus サービスプラットフォームで統合するサービスは、それぞれ独立した、業務上の意味を持つ単位で作成されたものです。Cosminexus サービスプラットフォームは、これらのサービスを、サービス間のメッセージの送受信（メッセージフロー）によって統合します。メッセージフローによって統合されたサービスでは、クライアントから一つのリクエストが送信されると、必要な複数のサービスを実行してそのリクエストにこたえます。これによって、クライアントから見たワンストップサービスを実現します。

Cosminexus サービスプラットフォームでは、統合した個々のサービスに接続するためのインタフェースを公開しています。クライアントは、公開されたインタフェースの情報を基に、サービスを利用するためのアプリケーションを作成できます。

なお、Cosminexus サービスプラットフォームで扱うサービスには、単体サービスと複合サービスがあります。複合サービスは、サービスを複数組み合わせることで実行順序を定義したものです。BPEL によってビジネスプロセスとして定義します。

Cosminexus サービスプラットフォームでのサービス統合の概要を、次の図に示します。

図 7-1 Cosminexus サービスプラットフォームでのサービス統合の概要



ユーザからのリクエストを受け付けた Cosminexus サービスプラットフォームは、リクエストの内容に応じて、登録されているサービスの中から適切なサービスを選び出します。同じサービスが複数登録されている場合は、それぞれのサービスの稼働状況や運用状態に合わせて呼び出し先を決定します。呼び出し先のサービスがビジネスプロセスである場合には、ビジネスプロセスのそれぞれの段階で必要なサービスを順番に呼び出して、処理を実行します。

(2) サービス統合の考え方

ここでは、サービス統合を実施する際の基本的な考え方について説明します。

サービスの作成単位

SOA の考え方に基づいたサービス統合を実現する場合、まず、サービスをどのような単位で構築するかを検討します。

SOA では、それぞれが独立しているサービスをつなぎ、必要に応じて組み替えることによって、新しいサービスを柔軟に作り出します。このため、組み替えの対象になるサービスの単位は、業務の変化が発生する単位に近く、かつ独立性が高いものにする必要があります。

7. サービスの統合を実現する

サービス統合の進め方

サービス統合の進め方には、一度にシステム全体を最適化する方法のほか、システム内で優先度が高い業務からサービス化を行い、段階を踏んで最適化を勤めていく方法があります。

レガシーシステムや ERP パッケージなどを使用して実現していた業務は、そのまま Cosminexus サービスプラットフォームのサービスとして利用できます。これによって、既存の資産を生かしたシステムの最適化が実現できます。詳細は、「7.1.2 レガシーシステムや ERP パッケージの統合」を参照してください。

また、新しいサービスを作成する場合は、対話型またはオンライン型のアプリケーションとして開発できます。対話型またはオンライン型のアプリケーションは、Cosminexus のアプリケーション開発環境を使用して、効率良く構築できます。詳細は、「6. アプリケーションを開発する」を参照してください。

7.1.2 レガシーシステムや ERP パッケージの統合

Cosminexus サービスプラットフォームでは、レガシーシステムや ERP パッケージと連携して、それぞれの機能をサービスとして統合できます。なお、Cosminexus サービスプラットフォームからサービスに接続するためには、アダプタというコンポーネントを使用します。

ここでは、Cosminexus サービスプラットフォームが連携できるレガシーシステムおよび ERP パッケージと、連携に使用するアダプタについて説明します。

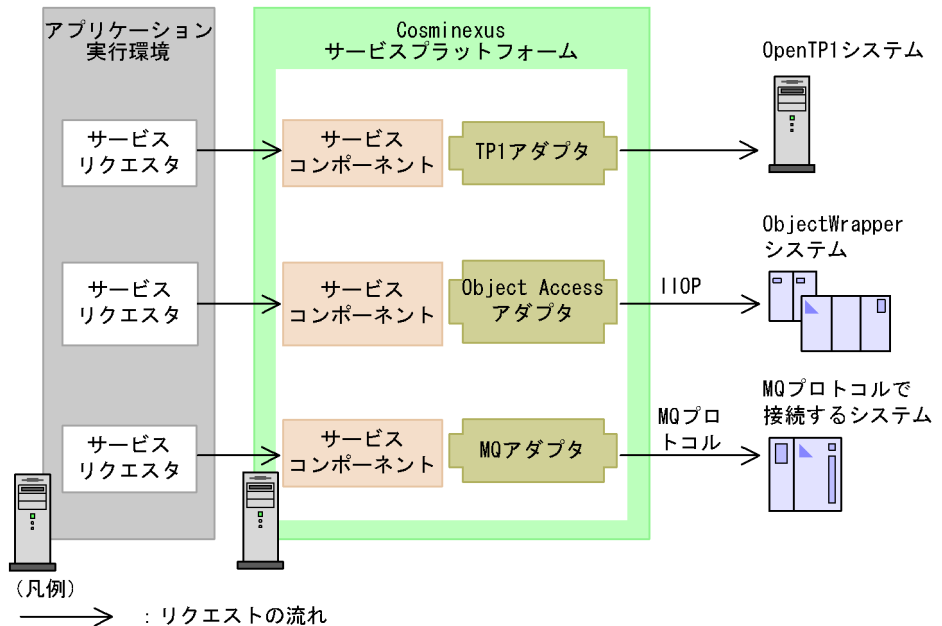
(1) レガシーシステムとの連携

Cosminexus サービスプラットフォームでは、次のレガシーシステムと連携できます。

- OpenTP1 システム
- Object Wrapper システム
- MQ プロトコルで接続するシステム

Cosminexus サービスプラットフォームとレガシーシステムとの連携について、次の図に示します。

図 7-2 レガシーシステムとの連携



それぞれのレガシーシステムについて説明します。なお、それぞれのレガシーシステムに接続するためのアダプタは、Cosminexus サービスプラットフォームとは別製品として提供されています。

OpenTP1 システム

Cosminexus サービスプラットフォームから、OpenTP1 システム内のサービス呼び出せます。

接続には、TP1 アダプタを使用します。

TP1 アダプタは、uCosminexus Service Adapter for TP1 で提供されるアダプタです。

参考

OpenTP1 システムから Cosminexus サービスプラットフォーム上のサービス呼び出す場合は、uCosminexus TP1 Gateway を使用します。

Object Wrapper システム

Cosminexus サービスプラットフォームから、Object Wrapper システム内のサービス呼び出せます。Object Wrapper システムとは、ホスト製品を CORBA オブジェクトとしてラッピングした、TPBroker システムの一つです。Object Wrapper システムとの通信は、TPBroker 経由で IIOP プロトコルによって実行されます。

接続には、Object Access アダプタを使用します。Object Access アダプタは、uCosminexus Service Adapter for Object Access で提供されるアダプタです。

7. サービスの統合を実現する

MQ プロトコルで接続するシステム

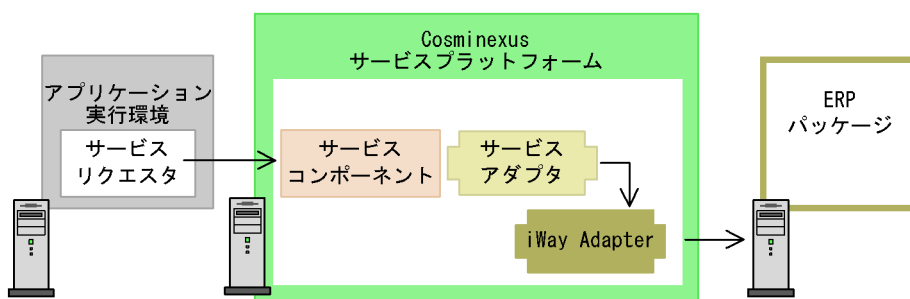
MQ プロトコルで接続可能な日立製品またはそれ以外の各種システムと連携できます。接続には、MQ アダプタを使用します。MQ アダプタは、uCosminexus TP1 Gateway で提供されるアダプタです。

(2) ERP パッケージとの連携

Cosminexus サービスプラットフォームでは、SAP R/3 や Oracle EBS などの ERP パッケージと連携できます。接続には、iWay Adapter を使用します。iWay Adapter は、ERP パッケージを Web サービスとして使用するためのアダプタです。このアダプタを使用することで、既存の ERP 資産を活用したシステム間連携が実現できます。

iWay Adapter を使用した ERP パッケージとの連携について、次の図に示します。

図 7-3 iWay Adapter を使用した ERP パッケージとの連携



(凡例)

→ : リクエストの流れ

iWay Adapter には、Cosminexus サービスプラットフォームが標準提供しているサービスアダプタ経由で接続します。

7.1.3 サービスコンポーネント開発の効率化

Cosminexus サービスプラットフォームの開発環境は、サービスコンポーネントの開発を効率良く進めるための機能を備えています。

ここでは、サービスコンポーネント開発の効率化を実現する機能について説明します。

(1) 開発環境の一括構築

Cosminexus サービスプラットフォームの開発環境は、一括構築できます。

一括構築では、J2EE サーバ、Hitachi Web Server、Cosminexus RM などを含めた、開発およびテストに必要な環境が、まとめて構築できます。これによって、前提環境の準備に時間を取られることなく、本来の目的である開発作業に速やかに着手できます。

(2) ビジネスモデリングツールで作成した BPEL 定義の利用

上流工程でのビジネスプロセスの設計に、ビジネスモデリングツールを使用している場合、これらのツールでエクスポートした BPEL 定義を開発環境にインポートして利用できます。作成したビジネスプロセス設計図を見ながら開発環境で再定義する必要はありません。

これによって、上流工程での成果を生かした、効率の良いサービスコンポーネント定義を実現できます。

(3) 操作性の高い GUI の提供

Eclipse を使用して、サービスコンポーネントをビジュアルな環境で開発できます。

また、GUI 間の連携によって、サービスコンポーネント定義の効率化が図れます。例えば、アダプタの定義で入力した情報をサービス呼び出しの定義で選択できます。

7.2 メッセージフローによってサービスを統合する

Cosminexus では、サービスの統合を Cosminexus サービスプラットフォームによって実現します。この節では、Cosminexus でサービス統合を実現する基盤を構成する三つの環境、およびそれらの構成について説明します。また、実行環境でのリクエストの流れについても説明します。

7.2.1 Cosminexus サービスプラットフォームを構成する三つの環境

Cosminexus サービスプラットフォームは、開発環境、運用環境、実行環境という三つの環境で構成されています。

Cosminexus サービスプラットフォームの開発環境または実行環境では、Developer Professional の開発環境の機能、または Application Server Enterprise の実行環境の機能に加えて、サービス統合を実現するための機能を使用できます。

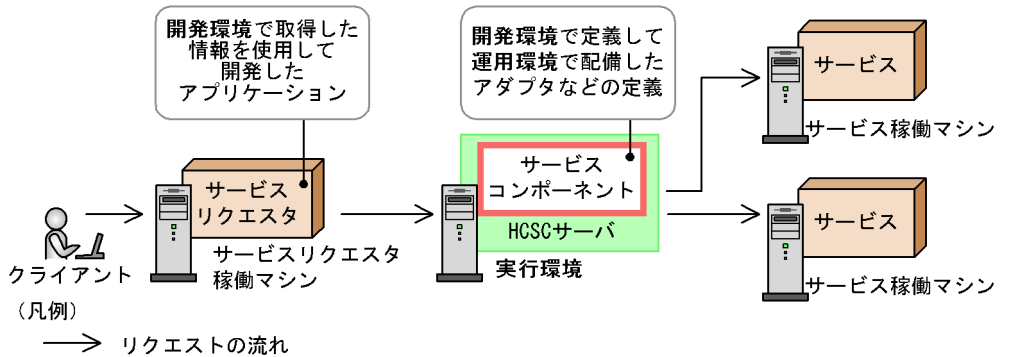
Cosminexus サービスプラットフォームでは、アプリケーションサーバやほかのシステム上で動作するサービスを統合して利用するために、アダプタを使用します。アダプタは、Cosminexus サービスプラットフォームがサービスと接続するために必要なコンポーネントです。Cosminexus サービスプラットフォームの開発環境で定義できます。Cosminexus サービスプラットフォームの実行環境では、クライアントからのリクエストに対応したアダプタを使用してサービスに接続し、そのサービスを利用します。アダプタなど、実行環境でサービスに接続したりサービスを利用したりするために必要な定義類を、サービスコンポーネントといいます。

サービスコンポーネントは、運用環境の機能を使用して、実行環境上に配備します。実行環境では、HCSC サーバが動作します。HCSC サーバは、Cosminexus のサービス統合の基盤である、ESB となるコンポーネントです。

また、サービスにリクエストを送信するアプリケーションに必要な情報は、Cosminexus サービスプラットフォームの開発環境で取得できます。これによって、クライアントアプリケーションの開発者は、実際のサービスがどのように実装されているかを意識しないでアプリケーションを開発できます。なお、Cosminexus サービスプラットフォームに対してリクエストを送信するアプリケーションを、サービスリクエスタといいます。

サービスを実行する流れと、サービスコンポーネント、HCSC サーバおよびサービスリクエスタの配置について、次の図に示します。

図 7-4 サービスを実行する流れと、サービスリクエスト、サービスコンポーネント、HCSC サーバ、およびサービスの配置



Cosminexus サービスプラットフォームを構成する三つの環境を、次の表に示します。

表 7-1 Cosminexus サービスプラットフォームを構成する三つの環境

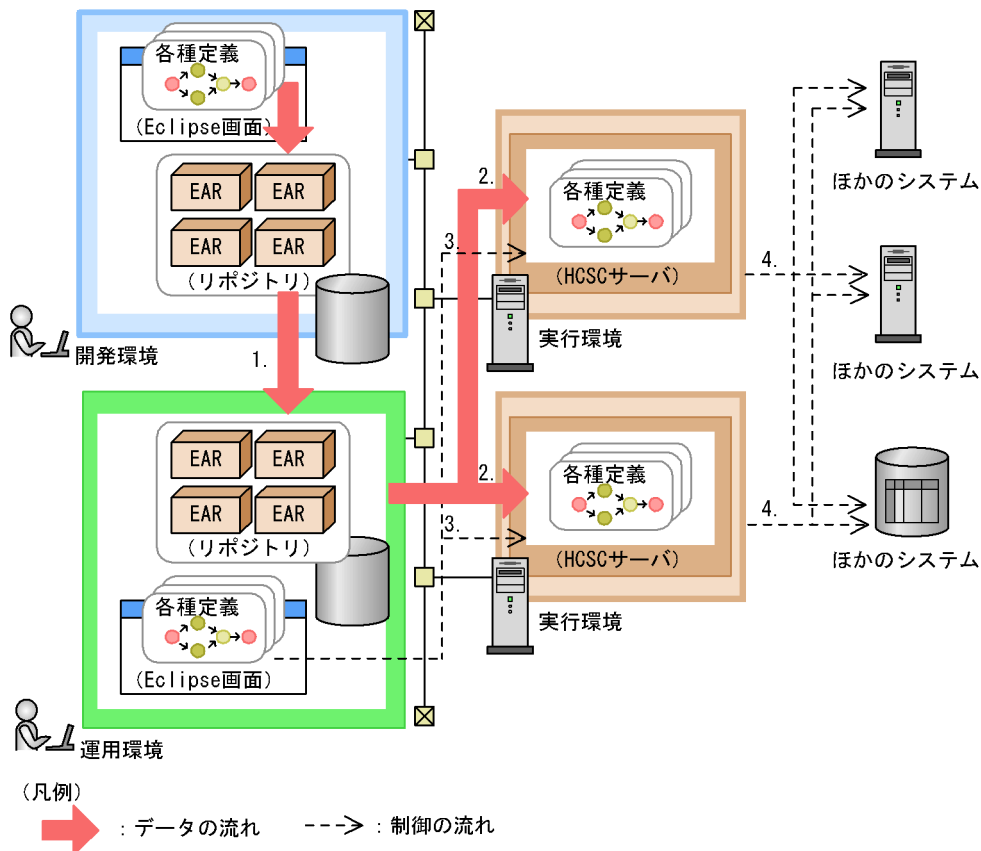
環境	概要
開発環境	サービス統合に必要な定義をするための環境です。サービスに接続するための情報や、ビジネスプロセスのロジックを、Eclipse のプラグインを使用して定義できます。また、定義した情報を EAR の形式にしてリポジトリに登録できます。
運用環境	開発環境で作成した EAR をリポジトリから読み込んで、実行環境に配備し、以降の運用操作を実行する環境です。また、実行環境から情報を収集して、サービスの稼働状態を確認できます。
実行環境	クライアントからの要求に応じてサービスやビジネスプロセスを呼び出して、業務を実行する環境です。HCSC サーバが動作しています。

これらの環境は、同じマシンに配置したり、異なるマシンに配置したりできます。

Cosminexus サービスプラットフォームのそれぞれの環境の関係を次の図に示します。また、それぞれの環境間でのデータおよび制御の流れについても示します。

7. サービスの統合を実現する

図 7-5 Cosminexus サービスプラットフォームのそれぞれの環境の関係



1. 開発環境から運用環境へのデータの流れ

開発環境では、Eclipseに組み込んだプラグインを使用して、サービス統合に必要なサービスやビジネスプロセスの定義、アダプタ定義、配備定義などのサービスコンポーネントを作成します。サービスコンポーネントは、EARの形式にアーカイブされ、開発環境のリポジトリに格納されます。

開発環境でリポジトリに格納されたEARファイルは、運用環境のリポジトリに取り込まれます。

2. 運用環境から実行環境へのデータの流れ

運用環境のEclipseに組み込んだプラグインを使用して、リポジトリの内容であるEARファイルを、実行環境に配備します。

3. 運用環境から実行環境への制御の流れ

デプロイしたサービスコンポーネントに対する操作や、実行環境でのサービスコンポーネントの運用状況の監視などを、運用環境のEclipseに組み込んだプラグインから実行します。

4. 実行環境からほかのシステムへの制御の流れ

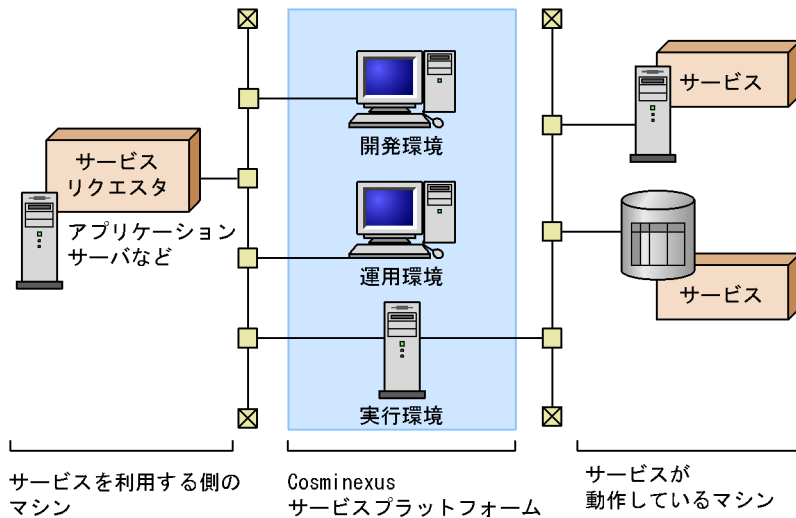
実行環境では、HCSC サーバが動作しています。サービスコンポーネントは、運用環境からの操作によって実行環境の HCSC サーバに配備されます。クライアントからのリクエストは、HCSC サーバ上のサービスコンポーネントを使用してほかのシステム上のサービス呼び出すことができます。

Cosminexus サービスプラットフォームの環境の詳細については、マニュアル「Cosminexus サービスプラットフォーム 概説」を参照してください。

7.2.2 Cosminexus サービスプラットフォームのシステム構成

Cosminexus サービスプラットフォームのシステム構成について説明します。次の図は、開発環境、運用環境、実行環境をすべて異なるマシンに配置したシステム構成の例です。

図 7-6 開発環境、運用環境、実行環境をすべて異なるマシンに配置したシステム構成



この図の場合、開発環境、運用環境、実行環境を分けて構築しています。

Cosminexus サービスプラットフォームを構成するそれぞれのマシンには、次の製品をインストールします。使用できる製品が複数ある場合は、どれかをインストールしてください。

表 7-2 Cosminexus サービスプラットフォームを構成するマシンにインストールする製品

構築する環境	使用できる製品
開発環境	Service Architect

7. サービスの統合を実現する

構築する環境	使用できる製品
運用環境	Service Platform uCosminexus Operator Service Architect
実行環境	Service Platform Service Architect

注 開発環境でのテスト用運用環境またはテスト用実行環境として使用できます。

なお、運用環境については、実行環境と同じマシンに構築することもできます。また、Cosminexus サービスプラットフォームの実行環境と、サービスが動作しているマシンを同じマシンにすることもできます。

ポイント

UNIX マシンに Cosminexus サービスプラットフォームの実行環境を構築する場合、実行環境と運用環境を同じマシンに構築できません。Eclipse に組み込んだプラグインからの運用操作は、Windows マシンで実行する必要があります。この場合は、運用環境を構築するマシンとして別途 Windows マシンを用意し、uCosminexus Operator をインストールしてください。

また、サービスを利用する側（サービスリクエスタ）のマシンには、次の製品をインストールします。サービスを利用する側のマシンは、クライアントの種類によって、インストールする製品が異なります。

サービスリクエスタが Web アプリケーションの場合

Web ブラウザからのリクエストを受け付けた Web アプリケーションから Cosminexus サービスプラットフォームにリクエストを送信する場合、Cosminexus サービスプラットフォームのクライアントは、アプリケーションサーバ、または任意の Java アプリケーション実行環境になります。アプリケーションサーバをクライアントにする場合は、次のどれかの製品をインストールしてください。

- Application Server Standard
- Application Server Enterprise
- Service Platform

サービスリクエスタが EJB クライアントアプリケーションの場合

EJB クライアントマシンから直接 Cosminexus サービスプラットフォームの実行環境にリクエストを送信することもできます。

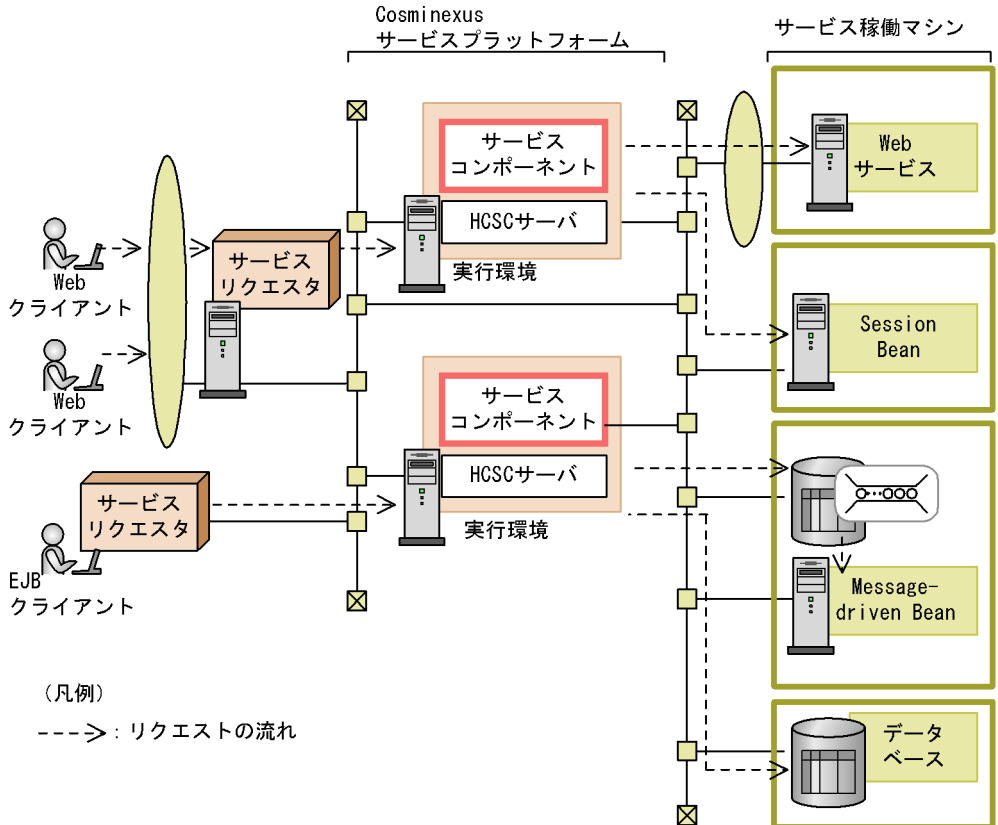
次の製品をインストールしてください。

- uCosminexus Client

7.2.3 サービスを利用する場合のリクエストの流れ

Cosminexus サービスプラットフォームで構築した実行環境でのリクエストの流れを次の図に示します。

図 7-7 Cosminexus サービスプラットフォームで構築した実行環境でのリクエストの流れ



Cosminexus サービスプラットフォームは、サービスリクエスタから受け付けたリクエストを、サービス内容やプロトコル種別に応じて各サービス稼働マシンに送信します。サービス稼働マシンで処理が実行された結果は、Cosminexus サービスプラットフォーム経由でサービスリクエスタに返却されます。なお、サービスリクエスタがビジネスプロセスを呼び出した場合は、サービス稼働マシンで動作する複数のサービスがビジネスプロセス定義に従って呼び出されます。

また、一つの実行環境にリクエストが集中するのを防ぐために、実行環境を冗長構成にして、リクエストを負荷分散することもできます。負荷分散は、利用するサービスが Web サービスまたは Session Bean の場合に実現できます。この場合、負荷分散には、サービスの形態に応じて、負荷分散機 (Web サービスの場合)、または Cosminexus の実行環境の機能である CTM (Session Bean の場合) を使用します。Cosminexus サービス

7. サービスの統合を実現する

プラットフォームでのリクエストの負荷分散については、マニュアル「Cosminexus サービスプラットフォーム 概説」を参照してください。

7.2.4 Cosminexus サービスプラットフォームの構築と運用

Cosminexus サービスプラットフォームの構築と運用については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。

ポイント

Service Architect を使用して Cosminexus サービスプラットフォームの開発環境またはデバッグ環境を構築する場合、開発・デバッグに必要な環境を一括構築できます。

7.3 メッセージフローによるサービス統合に必要なコンポーネントを定義 / 開発する

この節では、メッセージフローによるサービス統合を実現する場合に、サービスを利用するために必要なコンポーネントの定義および開発について説明します。

サービス統合を実現するためには、次のコンポーネントが必要です。

サービスコンポーネント

サービス統合基盤である HCSC サーバからサービスに接続するためのアダプタ定義や、複数のサービスを組み合わせて一つのサービスとして利用するためのビジネスプロセス定義などのことです。また、サービスを利用するためにデータ変換が必要な場合のデータ変換定義などの各種定義も含まれます。

サービスリクエスタ

HCSC サーバに対するサービスの呼び出し元になるプログラムです。同期処理または非同期処理を実行するためのプログラムを開発する必要があります。

Cosminexus サービスプラットフォームによるサービス統合に必要なコンポーネントの定義 / 開発については、マニュアル「Cosminexus サービスプラットフォーム 開発ガイド」を参照してください。

ここでは、概要を説明します。

7.3.1 サービスコンポーネントを定義する

Cosminexus サービスプラットフォームでサービスを統合するためには、サービスコンポーネントの定義が必要です。サービスコンポーネントの定義は、Cosminexus サービスプラットフォームの開発環境で実行します。開発環境では、次のようなサービスコンポーネントを定義できます。これらは、Eclipse に組み込んだプラグインを使用して、ビジュアルな環境で定義できます。

サービスやデータベースに接続するためのアダプタの定義

ビジネスプロセスの定義

データ変換の定義

配備定義

また、上流工程で、ビジネスモデリングツールを使用してビジネスプロセスを設計している場合、出力した BPEL 定義を開発環境にインポートして使用できます。

7.3.2 サービスリクエスタを開発する

Cosminexus サービスプラットフォームに対してリクエストを送信するサービスリクエ

7. サービスの統合を実現する

スタを開発します。なお、サービスリクエストは、任意の Java プログラム開発環境で開発できます。サービスリクエストは、利用するサービスの種類およびサービスへの接続に使用するプロトコルに応じて開発します。

Cosminexus サービスプラットフォームの開発環境では、サービスリクエストの開発に必要な、次の情報を取得できます。

表 7-3 サービスリクエスト開発時に利用できる情報

利用するサービス	通信プロトコル	取得できる情報	開発するクライアント
Web サービス (同期)	SOAP/HTTP	WSDL	SOAP クライアント
Session Bean (同期)	RMI-IIOP	スタブ	EJB クライアント
Message-driven Bean (非同期)	WS-Reliability	メッセージ送信先のキュー情報	メッセージを送信するクライアント
	Cosminexus RM 独自のプロトコル		

取得した情報を利用したアプリケーションの開発方法については、次のマニュアルを参照してください。

- WSDL を利用した SOAP アプリケーション開発
マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。
- スタブを利用したアプリケーション開発
マニュアル「Cosminexus アプリケーション開発ガイド」を参照してください。
- メッセージによる非同期通信をするアプリケーションの開発
マニュアル「Cosminexus Reliable Messaging」を参照してください。

参考

Web サービスを利用する場合、Cosminexus サービスプラットフォームでのサービスリクエストからの実行要求を受け付けるためのインターフェースとして、任意の形式を使用できます。詳細は、マニュアル「Cosminexus サービスプラットフォーム 概説」を参照してください。

8

ほかの製品との連携

この章では、Cosminexus とほかの製品との連携について説明します。

Cosminexus では、データベース、JP1 およびクラスタソフトウェアと連携したシステムの構築および運用ができます。

-
- 8.1 データベースとの連携
 - 8.2 JP1 との連携
 - 8.3 クラスタソフトウェアとの連携
-

8.1 データベースとの連携

この節では、Cosminexus と連携できるデータベースについて説明します。

Cosminexus は、次のデータベースと接続できます。

- HiRDB Version 8
- HiRDB Version 7
- XDM/RD E2
- Oracle10g R2
- Oracle10g
- Oracle9i R2
- Oracle9i
- SQL Server 2000
- SQL Server 2005

データベースごとに、使用できる機能が異なります。なお、SQL Server 2000 および SQL Server 2005 は、Windows の場合にだけ使用できます。

8.1.1 J2EE サーバから接続できるデータベース

ここでは、J2EE サーバから接続できるデータベースおよび接続に使用する JDBC ドライバについて説明します。データベースごとに接続に使用する JDBC ドライバが異なります。なお、データベースとの接続には、JDBC インタフェースを使用して接続する方法と、JDBC インタフェースに加えて JMS インタフェースを使用して接続する方法があります。

J2EE サーバから接続できるデータベース、JDBC ドライバおよび接続に使用するインタフェースの対応を、次の表に示します。

表 8-1 J2EE サーバから接続できるデータベース

データベース	JDBC ドライバ	JDBC インタフェース		JMS インタフェースおよび JDBC インタフェース	
		ローカルランザクション	グローバルランザクション	ローカルランザクション	グローバルランザクション
HiRDB Version 8	HiRDB Type 4 JDBC Driver				
	Cosminexus DABroker Library				

データベース	JDBC ドライバ	JDBC インタフェース		JMS インタフェースおよび JDBC インタフェース	
		ローカルトランザクション	グローバルトランザクション	ローカルトランザクション	グローバルトランザクション
HiRDB Version 7	Cosminexus DABroker Library				
XDM/RD E2	HiRDB Type 4 JDBC Driver		-	-	-
	Cosminexus DABroker Library (Database Connection Server)		-	-	-
Oracle10g R2 10.2.0	Oracle JDBC Thin Driver				
	Cosminexus DABroker Library			-	-
Oracle10g 10.1.0	Oracle JDBC Thin Driver				
	Cosminexus DABroker Library			-	-
Oracle9i R2 9.2.0	Oracle JDBC Thin Driver				
	Cosminexus DABroker Library			-	-
Oracle9i 9.0.1	Cosminexus DABroker Library		-	-	-
SQL Server 2000	SQL Server 2000 Driver for JDBC		-	-	-
SQL Server 2005	SQL Server 2005 JDBC Driver		-	-	-

(凡例)

- : 使用できます。かつ、使用を推奨します。
- : 使用できます。
- : 使用できません。

注

HiRDB クライアントが前提です。

8. ほかの製品との連携

接続できるデータベースおよび使用できる機能の詳細については、マニュアル「Cosminexus 機能解説」を参照してください。

8.1.2 Cosminexus サービスプラットフォームで接続できるデータベース

ここでは、Cosminexus サービスプラットフォームから接続できるデータベースおよび接続に使用する JDBC ドライバについて説明します。データベースごとに接続に使用する JDBC ドライバが異なります。なお、使用する機能によって接続できるデータベースおよび使用する JDBC ドライバが異なります。

Cosminexus サービスプラットフォームから接続できるデータベース、JDBC ドライバおよび接続に使用する機能の対応を、次の表に示します。

表 8-2 Cosminexus サービスプラットフォームから接続できるデータベース

データベース	JDBC ドライバ	メッセージング機能およびビジネスプロセス機能		DB アダプタ機能	
		ローカルランザクション	グローバルランザクション	ローカルランザクション	グローバルランザクション
HiRDB Version 8	HiRDB Type4 JDBC Driver				
	Cosminexus DABroker Library				
HiRDB Version 7	Cosminexus DABroker Library				
Oracle10g R2 10.2.0	Oracle JDBC Thin Driver				
	Cosminexus DABroker Library	-	-		
Oracle10g 10.1.0	Oracle JDBC Thin Driver				
	Cosminexus DABroker Library	-	-		
Oracle9i R2 9.2.0	Oracle JDBC Thin Driver				
	Cosminexus DABroker Library	-	-		

(凡例)

- : 使用できます。かつ、使用を推奨します。
- : 使用できます。
- : 使用できません。

接続できるデータベースおよび使用できる機能の詳細については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。

8.2 JP1 との連携

JP1 は、複合的な業務システムの統合運用管理を実現する、日立のミドルウェアです。Cosminexus のシステムと JP1 と連携することによって、高い運用性を持つシステムを構築・運用できます。

Cosminexus のシステムは、JP1 Version 8 および JP1 Version 7i と連携できます。

JP1 との連携で実現できる機能と使用する製品について、次の表に示します。

表 8-3 JP1 との連携で実現できる機能と使用する製品

機能	概要	JP1 Version 8 の製品	JP1 Version 7i の製品
障害の集中監視	システム全体を対象に、障害の集中監視ができます。	JP1/IM	JP1/IM
稼働性能の監視	システム全体を対象に、稼働性能の監視ができます。	JP1/PFM	-
システム構成定義と管理	システム構築時に使用するシステムの構成情報を定義したり、定義したシステムの構成情報を収集したりできます。	JP1/IM - CM	JP1/IM - CM
ジョブによる運用の自動化	Cosminexus で管理しているサーバやプロセスの起動/停止を JP1/AJS2 のジョブを使用して自動化できます。	JP1/AJS2	JP1/AJS2
シナリオによる運用の自動化	シナリオを使用してシステムの自動運転ができます。	JP1/AJS2 - SO	JP1/AJS2 - SO
SNMP での稼働情報の取得	SNMP で稼働情報を取得します。UNIX の場合だけ使用できます。	JP1/Cm2/ESA	JP1/Cm2/ESA
監査ログの収集と一元管理	Cosminexus など、システム内の製品が出力した監査ログを収集して一元管理します。	JP1/NETM/ Audit	-

(凡例)

- : 該当しません。

注 Smart Composer 機能を使用してシステムを構築している場合に実現できます。

それぞれの機能の詳細については、マニュアル「Cosminexus 機能解説」を参照してください。

JP1 Version 8 の製品および JP1 Version 7i の製品の詳細については、それぞれの製品のマニュアルを参照してください。

8.3 クラスタソフトウェアとの連携

Cosminexus で構築したシステムは、クラスタソフトウェアと連携させることによって、可用性を高めた運用を実現できます。

クラスタソフトウェアと連携して Cosminexus システムを運用することで、アプリケーションサーバに障害が発生したときに、待機させておいたアプリケーションサーバに切り替えたり、障害が発生したアプリケーションサーバのリカバリ処理を待機しているリカバリサーバで実施したりできます。また、運用管理用のサーバに障害が発生したときにも、待機させておいたサーバに切り替えることができます。

Cosminexus が連携できるクラスタソフトウェアを次の表に示します。

表 8-4 連携できるクラスタソフトウェア

Cosminexus が動作する OS	Microsoft Cluster Service	HA モニタ
Windows		-
AIX	-	
HP-UX	-	
Linux	-	
Solaris	-	-

(凡例) : 使用できます。 - : 使用できません。

注 Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 (x64) および Windows Server 2003 R2 (x64) が該当します。

クラスタソフトウェアと連携して実現できる機能の詳細については、マニュアル「Cosminexus 機能解説」を参照してください。

9

Cosminexus のマニュアル体系

この章では、Cosminexus の実行環境、開発環境およびサービス統合環境に対応するマニュアルの体系について説明します。

9.1 マニュアル体系

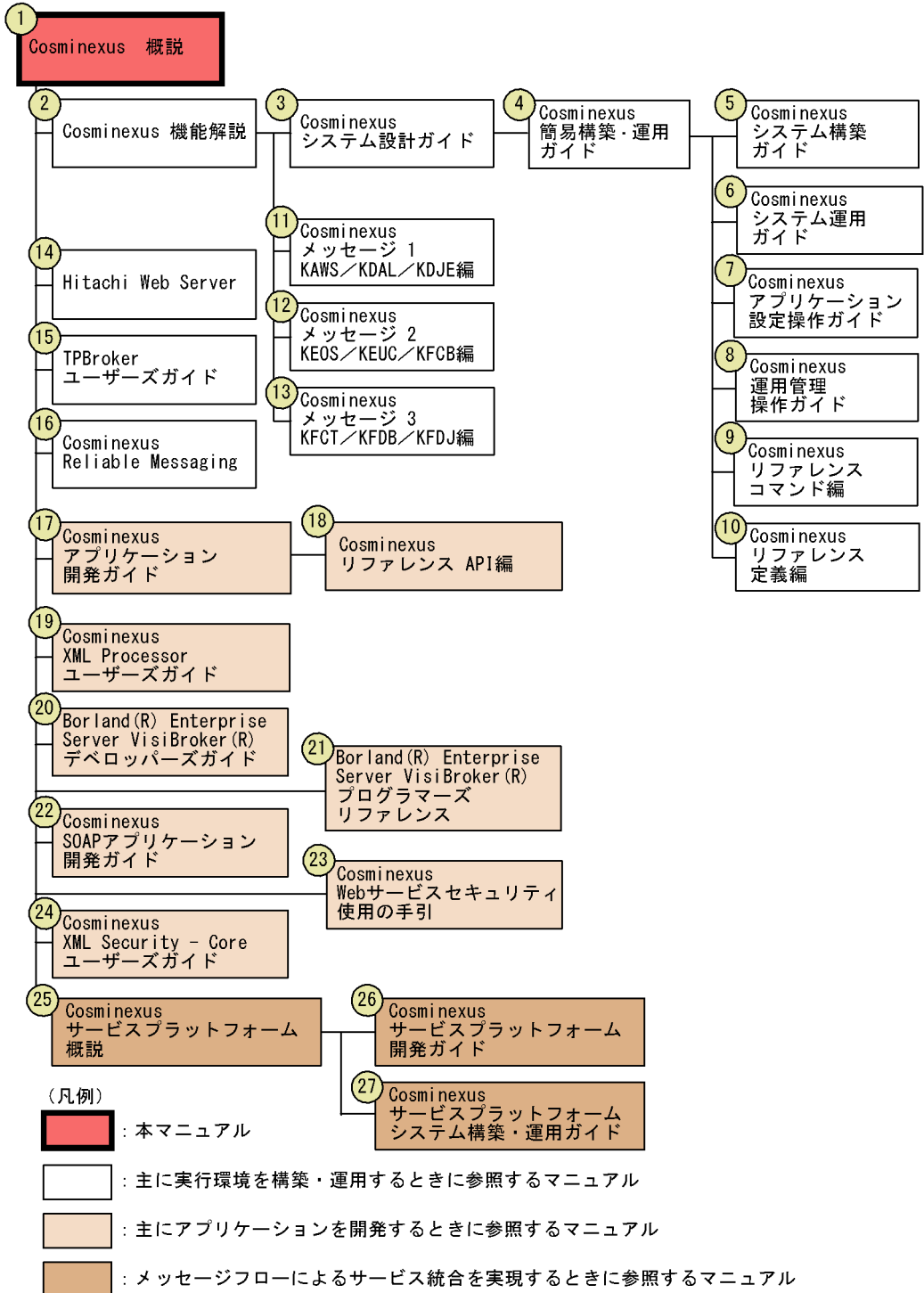
9.2 マニュアルの読書手順

9.1 マニュアル体系

Cosminexus の実行環境，開発環境およびサービス統合環境に対応するマニュアルについて説明します。

Cosminexus の実行環境および開発環境に対応するマニュアルのマニュアル体系を，次の図に示します。

図 9-1 Cosminexus のマニュアル体系（実行環境および開発環境に対応するマニュアル）



9. Cosminexus のマニュアル体系

それぞれのマニュアルで説明している内容の概要を次の表に示します。なお、表の項番は図 9-1 の項番と対応しています。

表 9-1 Cosminexus のマニュアル概要

項番	マニュアル名	概要
1	Cosminexus 概説	Cosminexus の製品概要について説明しています。このマニュアルです。
2	Cosminexus 機能解説	Cosminexus の実行環境で提供する機能について説明しています。
3	Cosminexus システム設計ガイド	システム設計時に、システムの目的に応じたシステム構成や運用方法を検討するための指針について説明しています。また、チューニングの方法についても説明しています。
4	Cosminexus 簡易構築・運用ガイド	Smart Composer 機能およびセットアップウィザードを使用して、システムを構築・運用する手順について説明しています。また、Smart Composer 機能およびセットアップウィザードが提供するコマンドやファイルについても説明しています。
5	Cosminexus システム構築ガイド	システム構築時に必要な機能の設定方法について説明しています。
6	Cosminexus システム運用ガイド	Cosminexus を使用したシステムの運用方法のうち、Smart Composer 機能を使用する運用以外の方法について説明しています。トラブルシューティングについても説明しています。
7	Cosminexus アプリケーション設定操作ガイド	サーバ管理コマンド、および Server Plug-in を使用したアプリケーションおよびリソースの操作について説明しています。
8	Cosminexus 運用管理操作ガイド	運用管理ポータル画面および画面操作について説明しています。
9	Cosminexus リファレンス コマンド編	システムを構築・運用するときに使用するコマンドについて説明しています。
10	Cosminexus リファレンス 定義編	システムを構築・運用するとき、またはアプリケーションを開発するときに、使用するファイルの形式について説明しています。
11	Cosminexus メッセージ 1 KAWS / KDAL / KDJE 編	KAWS, KDAL, または KDJE で始まるメッセージについて説明しています。
12	Cosminexus メッセージ 2 KEOS / KEUC / KFCB 編	KEOS, KEUC, または KFCB で始まるメッセージについて説明しています。
13	Cosminexus メッセージ 3 KFCT / KFDB / KFDJ 編	KFCT, KFDB, または KFDJ で始まるメッセージについて説明しています。
14	Hitachi Web Server	Hitachi Web Server (Web サーバ) の構築、管理方法について説明しています。
15	TPBroker ユーザーズガイド	Cosminexus TPBroker の概要、機能、運用方法について説明しています。

項番	マニュアル名	概要
16	Cosminexus Reliable Messaging	Cosminexus RM を使用したメッセージの非同期通信によるアプリケーションの連携方法について説明しています。
17	Cosminexus アプリケーション開発ガイド	構築したシステムで動作させる、アプリケーションの開発方法について説明しています。また、開発環境のセットアップ方法についても説明しています。
18	Cosminexus リファレンス API 編	アプリケーションの開発で使用する API およびタグについて説明しています。
19	Cosminexus XML Processor ユーザーズガイド	Cosminexus XML Processor が提供する XML パーサー・XSLT トランスフォーマの機能、作成方法、および使用方法について説明しています。
20	Borland(R) Enterprise Server VisiBroker(R) デベロッパーズガイド	Borland Enterprise Server VisiBroker の基本的な使用方法および高度な機能の取り扱い方法について説明しています。
21	Borland(R) Enterprise Server VisiBroker(R) プログラマーズリファレンス	Borland Enterprise Server VisiBroker が提供しているクラスとインタフェースの情報、プログラマーツール、およびコマンドラインオプションについて説明しています。
22	Cosminexus SOAP アプリケーション開発ガイド	Cosminexus が提供する機能を利用して、SOAP アプリケーションを開発、実行する方法について説明しています。
23	Cosminexus Web サービスセキュリティ 使用の手引	Web サービスセキュリティ機能について説明しています。
24	Cosminexus XML Security - Core ユーザーズガイド	XML 署名データの生成・検証機能、および XML 暗号化・復号化機能について説明しています。
25	Cosminexus サービスプラットフォーム 概説	Cosminexus サービスプラットフォームの特長、機能、およびシステムの開発・運用方法の概要について説明しています。
26	Cosminexus サービスプラットフォーム 開発ガイド	Cosminexus サービスプラットフォームを利用したシステムの開発について説明しています。
27	Cosminexus サービスプラットフォーム システム構築・運用ガイド	Cosminexus サービスプラットフォームを利用したシステムの構築および運用方法について説明しています。

注 マニュアル「uCosminexus UDDI Registry システム運用ガイド」は、Application Server、Developer および Cosminexus サービスプラットフォームとは異なる製品に添付されるマニュアルであるため、ここでは説明しません。また、マニュアル「Cosminexus Application Designer 操作ガイド」についての説明も、ここでは省略します。

9.2 マニュアルの読書手順

この節では、Cosminexus のマニュアルの読書手順について説明します。

9.2.1 実行環境を構築・運用する場合の読書手順

ここでは、実行環境を構築する場合の読書手順について説明します。なお、実行環境の種類によって、参照するマニュアルが異なります。

実行環境を構築・運用する場合に参照するマニュアルの読書手順を次の図に示します。

表 9-2 実行環境を構築・運用する場合に参照するマニュアルと目的

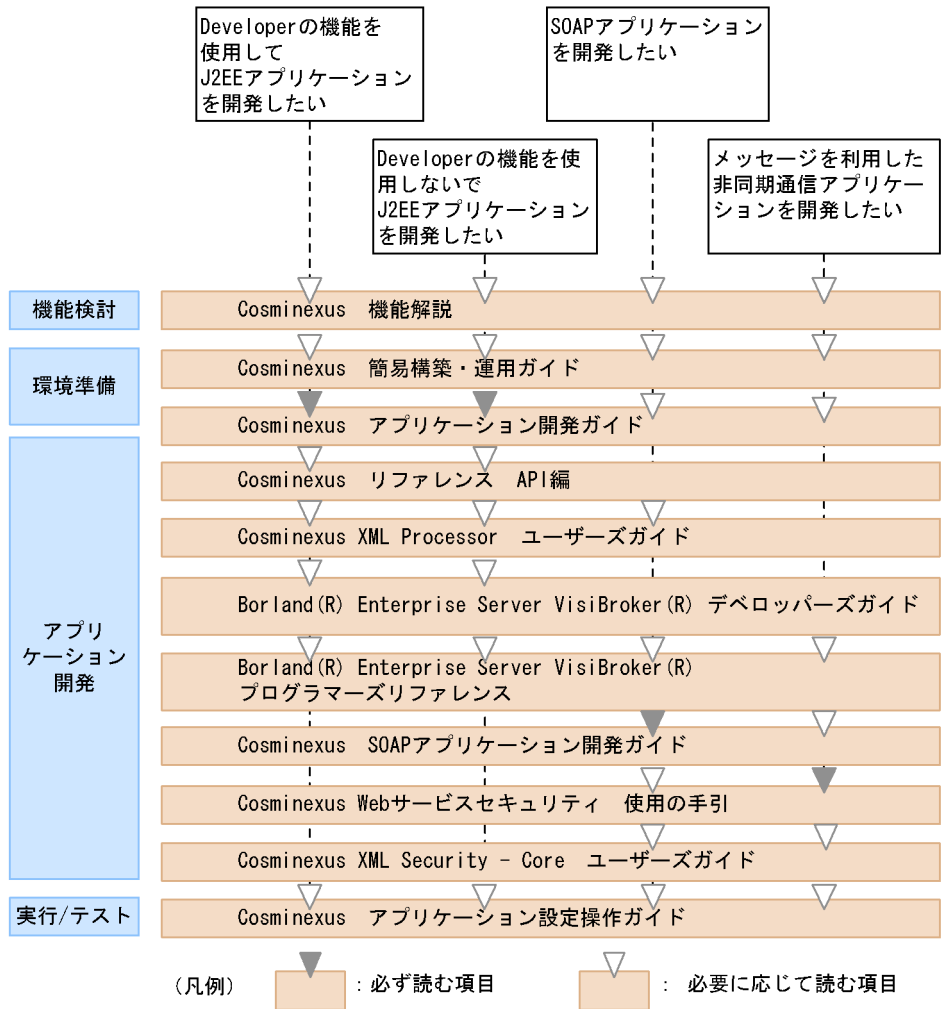
マニュアル	参照する目的
Cosminexus 機能解説	実行環境で使用できる機能を理解して、システムで使用する機能を検討するために参照します。
Cosminexus システム設計ガイド	実行環境のシステム設計をするために参照します。 使用する機能に応じて、実行環境をどのような構成で構築するかを検討するか、また、構成に応じてシステム構築をどの手順で実行するかを検討するために参照します。そのほか、セキュリティについての情報、チューニングについての情報についても参照します。
Cosminexus 簡易構築・運用ガイド	実行環境を Cosminexus の推奨構成で構築する場合に、Smart Composer 機能またはセットアップウィザードを使用してシステムを構築・運用するために参照します。
Cosminexus システム構築ガイド	実行環境を拡張構成などの任意の構成で構築する場合に、システムを構築するために参照します。
Cosminexus システム運用ガイド	実行環境を拡張構成などの任意の構成で構築する場合に、システムを運用するために参照します。
Cosminexus アプリケーション設定操作ガイド	サーバ管理コマンド、および Server Plug-in を使用する場合に、操作手順を確認するために参照します。
Cosminexus 運用管理操作ガイド	運用管理ポータルを使用してシステムを構築・運用する場合に、運用管理ポータルの使用方法について参照します。
Cosminexus リファレンス コマンド編	Cosminexus のシステムを構築・運用する場合に、コマンドを使用するときに参照します。
Cosminexus リファレンス 定義編	Cosminexus のシステムを構築・運用する場合、またはアプリケーションを開発する場合に、ファイルを使用するときに参照します。
Hitachi Web Server	Web サーバとして Hitachi Web Server を使用する場合に Web サーバの構築・運用をするために参照します。
TPBroker ユーザーズガイド	Cosminexus TPBroker のログについて参照したい場合などに、必要に応じて参照します。
Cosminexus Reliable Messaging	リソースアダプタとして Cosminexus RM を使用して、メッセージを使用した非同期通信をする環境を構築・運用する場合に参照します。
Cosminexus SOAP アプリケーション開発ガイド	実行環境で SOAP アプリケーションを実行する場合に参照します。
Cosminexus メッセージ 1 KAWS / KDAL / KDJE 編	KAWS, KDAL, または KDJE で始まるメッセージが出力された場合に参照します。
Cosminexus メッセージ 2 KEOS / KEUC / KFCB 編	KEOS, KEUC, または KFCB で始まるメッセージが出力された場合に参照します。
Cosminexus メッセージ 3 KFCT / KFDB / KFDJ 編	KFCT, KFDB, または KFDJ で始まるメッセージが出力された場合に参照します。

9.2.2 アプリケーションを開発する場合の読書手順

ここでは、アプリケーションを開発する場合の読書手順について説明します。

アプリケーションを開発する場合に参照するマニュアルの読書手順を次の図に示します。

図 9-3 アプリケーションを開発する場合に参照するマニュアルの読書手順



それぞれのマニュアルを参照する目的を次に示します。

表 9-3 アプリケーションを開発する場合に参照するマニュアルと目的

マニュアル	参照する目的
Cosminexus 機能解説	実行環境で使用できる機能、およびその実装方法を理解するために参照します。アプリケーションで実現したい機能について知りたい場合に必要に応じて参照してください。

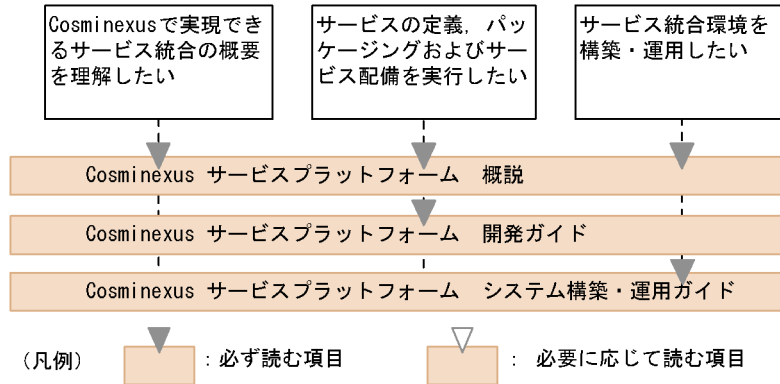
9. Cosminexus のマニュアル体系

マニュアル	参照する目的
Cosminexus 簡易構築・運用ガイド	開発の前提になる環境を一括構築する方法を知りたい場合に参照します。
Cosminexus アプリケーション開発ガイド	Cosminexus でのアプリケーション開発の概要，およびアプリケーション開発手順について理解するために参照します。
Cosminexus リファレンス API 編	アプリケーション実装時に Cosminexus が提供している API を使用する場合に参照します。
Cosminexus XML Processor ユーザーズガイド	JAXP を使用したアプリケーションを実装したい場合に必要に応じて参照します。
Borland(R) Enterprise Server VisiBroker(R) デベロッパーズガイド	J2EE 環境で動作する CORBA クライアントアプリケーションを開発する場合に必要に応じて参照します。
Borland(R) Enterprise Server VisiBroker(R) プログラマーズリファレンス	J2EE 環境で動作する CORBA クライアントアプリケーションを開発する場合に必要に応じて参照します。
Cosminexus SOAP アプリケーション開発ガイド	Web サービスを使用するための SOAP アプリケーションを開発する場合に参照します。
Cosminexus Web サービスセキュリティ 使用の手引	SOAP アプリケーションで Web サービスセキュリティの技術を使用する場合に参照します。
Cosminexus XML Security - Core ユーザーズガイド	アプリケーションで XML 署名や XML 暗号の技術を使用する場合に参照します。
Cosminexus アプリケーション設定操作ガイド	開発したアプリケーションをテストする場合に必要に応じて参照します。

9.2.3 サービス統合環境を構築・運用する場合のマニュアルの読書手順

ここでは，メッセージフローによるサービス統合環境を構築・運用する場合に参照するマニュアルの読書手順を次の図に示します。

図 9-4 メッセージフローによるサービス統合環境を構築・運用する場合に参照するマニュアルと目的



それぞれのマニュアルを参照する目的を次に示します。

表 9-4 メッセージフローによるサービス統合環境を構築・運用する場合のマニュアルを参照する目的

マニュアル	参照する目的
Cosminexus サービスプラットフォーム 概説	Cosminexus で構築できるサービス統合環境について知りたい場合に参照します。Cosminexus サービスプラットフォームの開発環境、運用環境および実行環境を合わせた全体像を理解したい場合に参照します。
Cosminexus サービスプラットフォーム 開発ガイド	Cosminexus サービスプラットフォームの開発環境でのシステム開発手順について知りたい場合に参照します。開発環境では、サービスの定義、パッケージング、サービス配備の定義などを実行します。
Cosminexus サービスプラットフォーム システム構築・運用ガイド	開発環境で定義したサービス定義を実行環境に配備して運用したい場合に参照します。サービス統合環境の構築についても説明しています。

付録

付録 A Cosminexus が対応する標準仕様

付録 B 用語解説

付録 A Cosminexus が対応する標準仕様

ここでは、Cosminexus が対応している標準仕様について説明します。

J2EE の標準仕様

Cosminexus は、J2EE 1.4 に対応しています。サポートしている API のバージョンの詳細については、マニュアル「Cosminexus 機能解説」の、サーバの動作モードの説明を参照してください。

XML, Web サービス関連の標準仕様

次の仕様に対応しています。

- SOAP1.1
- WSDL1.1
- UDDI2.0
- JAXP1.3
- JAXR1.0
- WS-I Basic Profile 1.0a
- Web Services Security : SOAP Message Security Draft 17
- WS-Reliability1.1

SSL 関連の標準仕様

- SSL バージョン 2, SSL バージョン 3
- TLS バージョン 1

OMG 分散オブジェクト関連の標準仕様

- CORBA 2.5
- CORBA Object Transaction Service 1.3

付録 B 用語解説

Cosminexus で使用する用語について説明します。

(英字)

Application Server

Cosminexus の実行環境を構築する基盤製品です。Application Server Standard と、Application Server Enterprise の総称です。

Cosminexus

アプリケーションサーバを中核とした、性能および信頼性の高い J2EE アプリケーションを実行および開発するためのシステム構築基盤製品です。

Cosminexus Component Container

サーバ・サイドの業務処理プログラム（ビジネスロジック）をコンポーネントとして実行するための構成ソフトウェアです。

Cosminexus Component Transaction Monitor

クライアントからのリクエストのスケジューリングを実現する構成ソフトウェアです。

Cosminexus Manager

Cosminexus のシステムの運用管理機能です。

Cosminexus Performance Tracer

リクエストが処理されるときに、決められたポイントごとに各機能が出力する性能解析情報をファイルに出力するための構成ソフトウェアです。

Cosminexus Reliable Messaging

Cosminexus システム上の J2EE アプリケーションがメッセージを使用して非同期に通信するための構成ソフトウェアです。JMS インタフェースでのメッセージ通信機能を J2EE アプリケーションに提供します。

Cosminexus Service Coordinator

Cosminexus によってサービス統合を実現するための基盤機能（Cosminexus サービスプラットフォーム）を提供する構成ソフトウェアです。SOA の概念でシステム内のサービスを統合するための、メッセージング、ビジネスプロセス、データ変換、DB アダプタなどの機能を提供します。

Cosminexus サービスプラットフォーム

SOA を実現するシステムの開発・運用の基盤となる製品です。Service Platform と Service Architect の総称です。SOA のメリットを引き出して、柔軟性と信頼性の高いシステムを開発および運用できます。

CTM

Cosminexus Component Transaction Monitor のことです。リクエストのスケジューリングをするための機能を提供しています。

DB Connector

データベースに接続するためのリソースアダプタです。

DB アダプタ

サービス統合環境で、データベースの操作をサービス部品として利用する場合に、操作するデータベースや実行する SQL を定義したサービスコンポーネントです。

Developer

Cosminexus の開発環境を提供する製品です。アプリケーションの開発からデバッグまでを支援します。Developer Standard および Developer Professional の総称です。

Eclipse

Eclipse プロジェクト (eclipse.org) が提供するオープンソースの統合開発環境です。ソースコードの編集支援機能やデバッグ機能など、アプリケーションの開発効率を向上させる各種機能を備えています。

Eclipse プラグイン

Developer が提供する Eclipse 用のプラグインです。このプラグインを Eclipse に組み込むことで、Eclipse からアプリケーション開発ツールを起動したり、開発したアプリケーションを実行したりできます。

EIS

データベースやトランザクションサーバなど、企業内に構築されているバックエンドシステムです。

EJB

Enterprise JavaBeans の略です。業務ロジックをプログラムとして記述したビジネスロジック、および永続的データを格納するオブジェクトであるエンティティを Java コンポーネント化したものです。Sun Microsystems, Inc. から EJB 仕様が公開されています。

EJB クライアント

J2EE サーバ上で開始されている Enterprise Bean を呼び出すクライアントプログラムです。次の 3 種類があります。

- EJB クライアントアプリケーション
- サブレットまたは JSP などの Web アプリケーション
- ほかの Enterprise Bean

EJB クライアントアプリケーション

Enterprise Bean を呼び出す Java アプリケーションです。

EJB コンテナ

Enterprise Bean を制御すると同時に、通信、トランザクション管理などのシステムレベルのサービスを提供する実行環境です。Enterprise Bean の実体は、EJB コンテナの中で実行されます。

Enterprise Bean

ビジネスロジックを EJB アーキテクチャに従って作成したものです。業務処理プログラムに該当します。

J2EE

Java™ 2 Platform, Enterprise Edition の略です。Sun Microsystems, Inc. から J2EE 仕様が公開されています。

J2EE アプリケーション

JSP, サブレット, Enterprise Bean など構成されるアプリケーションです。Cosminexus で扱う J2EE アプリケーションの形式には, EAR ファイル形式でパッケージ化されたアーカイブ形式のアプリケーションと, アーカイブ化しない展開ディレクトリ形式のアプリケーションがあります。EAR ファイル形式でパッケージ化されたアプリケーションの場合, 複数の EJB-JAR ファイル, 複数の WAR ファイル, および一つの DD から構成されます。

J2EE サーバ

J2EE コンテナを生成, 実行する環境です。

Java

1991 年に Sun Microsystems, Inc. 内の研究グループが開発したインタプリタ言語を基に, C++ をより移植性と信頼性が高くなるように改良した言語です。Web の普及によってブラウザに应用されています。

JavaVM

Java Virtual Machine の略です。Java 言語で作られたプログラムを実行するためのソフトウェアです。

Cosminexus が使用する JavaVM は, 構成ソフトウェアである Cosminexus Developer's Kit for Java によって提供される, 独自の JavaVM です。

JP1

日立の統合システム運用管理ソフトウェアです。

JSP

JavaServer Pages の略です。Web ページを作成する HTML の中に JSP 固有のタグを埋め込んで Enterprise Bean を呼び出すことや, サブレットから値を受け取ることができます。また, 簡単な Java 言語を直接記述することもできます。

Management Server

Cosminexus の運用管理を実現するサーバプログラムです。複数のマシンに配置されたアプリケーションサーバの運用管理を一括で実行できます。

Message-driven Bean

JMS と連携するメッセージ駆動タイプの Bean です。EJB コンテナは JMS の Destination からの JMS メッセージ受信を契機に Bean を起動します。

MVC

Model と View, そして Controller の頭文字を持ったものです。Model は, システムの中でロジック

クを担当します。View は、表示、入出力といった部分を担当します。Controller は、View と Model を制御します。

MyEclipse

Java アプリケーションの開発環境です。Eclipse でアプリケーションを開発するときに使用するプラグインがまとめられています。MyEclipse を使用するとアプリケーションの作成、J2EE サーバへのアプリケーションのデプロイ、J2EE サーバの起動・停止などができます。また、Struts や JSF などのフレームワークを使用したアプリケーションの開発や、UML によるアプリケーション設計もできます。

Cosminexus では、構成ソフトウェアとして、MyEclipse の日本語版環境である MyEclipse for Cosminexus を提供しています。

RMI-IIOP

JavaRMI と JavaIDL を統合した API です。

Session Bean

主に業務処理を実行するための Bean です。Session Bean は、クライアントを一つだけ持つことができます。クライアントが終了すると、対応する Session Bean も終了します。クライアントとのセッションの間だけ動作するため、一時的であり、非永続的な Bean です。Session Bean は、Stateful Session Bean と、Stateless Session Bean に分類されます。

Smart Composer 機能

Cosminexus を使用してアプリケーションサーバを構築する場合に、一般的な 3 階層モデルのアプリケーションサーバのシステムを、一括で構築および運用できるよう支援する機能です。システム全体に対して、システムの設定、および J2EE アプリケーションやリソースアダプタのデプロイを一括で実行できるので、簡単、迅速にシステム構築ができます。構築したシステムを一括で起動したり、一括でシステムの設定を変更したりする運用機能についてもサポートしています。

SOA

Service Oriented Architecture (サービス指向アーキテクチャ) の略です。サービスという概念を基本としてシステムを構築します。アプリケーションなどによって提供される業務処理をサービスとして扱います。既存のアプリケーションや新しく開発したアプリケーションを組み合わせ、要件に応じたサービスを柔軟に提供する業務システムを構築できます。

SOAP

Simple Object Access Protocol (簡易オブジェクト・アクセス・プロトコル) の略です。ネットワーク上のサービスを呼び出す際のメッセージを規定したもので、HTTP と XML がベースとなっています。

SOAP アプリケーション

Web の標準技術を利用してネットワーク上でサービスを公開、実行できるアプリケーションです。

uCosminexus Client

EJB クライアント環境を構築するための製品です。

uCosminexus Operator

Application Server Standard , Application Server Enterprise または Service Platform をインス

インストールした実行環境のマシンとは別の Windows マシンから、GUI (Eclipse のプラグイン) を使用して実行環境を操作するための製品です。

uCosminexus Web Redirector

実行環境を構築する場合に、Web サーバとアプリケーションサーバを異なるマシンに配置するときに使用する製品です。

Web アプリケーション

Web ブラウザを備えたクライアントを対象に作成されたアプリケーションです。具体的には、サーブレットプログラム、JSP ページ、HTML/XML ドキュメントなどの集合体です。

Web コンテナ

J2EE アーキテクチャの Web コンポーネント規約を実装するコンテナです。セキュリティ、トランザクションなどの各種サービスを提供する実行環境です。Web アプリケーションは、Web コンテナ上で動作します。

Java Servlet2.4 仕様、および JavaServer Pages Specification v2.0 仕様に準拠した Web アプリケーションを実行できます。

Web サーバ

Web ブラウザからのリクエスト受信および Web ブラウザへのデータ送信に関連する処理を実行するプログラムです。Cosminexus では、Hitachi Web Server、Microsoft IIS、またはインプロセス HTTP サーバを使用できます。インプロセス HTTP サーバは、J2EE サーバプロセス内で動作する Web サーバです。

Web サービス

インターネット上に分散するアプリケーションが提供するサービスを、クライアントに API として提供するサービスです。

WS-Reliability

標準化団体 OASIS で標準化された、インターネット向けのオープンな非同期通信プロトコルです。

(ア行)

アプリケーションサーバ

情報システムの間頭に位置し、ユーザの要求 (プレゼンテーション層) とデータベースなどの業務システム (データ層) の処理を橋渡しするためのアプリケーション層を構築するためのミドルウェアです。日立のアプリケーションサーバ Cosminexus は、業務の開発から運用まで一貫した環境を提供します。

(カ行)

業務アプリケーション

業務単位の J2EE アプリケーションや Web アプリケーション、バッチアプリケーションなどのことです。

(サ行)

サービス

SOA の概念で扱うアプリケーションやデータなどの単位です。一つの J2EE アプリケーションや複数の J2EE アプリケーションなどで実現する業務処理に対応します。SOA では、サービスの組み合わせによって業務システムを構築します。

サービスコンポーネント

Cosminexus サービスプラットフォームで使用する、サービスに接続するためのアダプタ定義やビジネスプロセス定義などの総称です。

サービスユニット

J2EE サーバや Web サーバで構成される業務サービスを提供する、閉じた部分系のことです。Smart Composer 機能では、サービスユニット単位でシステムを構築したり、一括起動や一括停止をしたりできます。Smart Composer 機能を利用してシステムを構築するときの概念です。

サービスリクエスト

Cosminexus サービスプラットフォームのサービスを利用する、サービスの呼び出し元になるプログラムです。Cosminexus サービスプラットフォームが提供するサービスに関する情報を使用して開発します。

サーブレット

サーバ側で Java を実行させる方法の一つです。サーブレットは、Web サーバに対して、単に HTML 文書や画像ファイルを送るだけではなく、Web サーバと連携して、アプリケーションを実行し、その結果を HTML 文書として送り返す機能を提供します。

スケールアウト

システム全体の処理性能を向上させることを目的として、サーバの台数を増やすことをいいます。

スケールイン

システムの規模を縮小する場合などに、サーバの台数を減らすことをいいます。

(夕行)

デプロイ

J2EE アプリケーションの場合、J2EE サーバ内にインポートした J2EE アプリケーションを、クライアントから実行可能な状態にすることです。J2EE リソースアダプタの場合、J2EE サーバ内にインポートした J2EE リソースアダプタを、その J2EE サーバ上で動作するすべての J2EE アプリケーションから使用可能な状態にすることです。

(八行)

バッチアプリケーション

バッチ処理を実装した Java アプリケーションです。バッチ実行コマンドを使用して、バッチサーバ上で実行します。また、バッチ実行コマンドを JP1 のジョブとして定義しておくこと、JP1/AJS2 からバッチアプリケーションを実行できます。

バッチサーバ

バッチアプリケーションを実行するためのサーバプロセスです。バッチアプリケーションを実行できます。また、バッチアプリケーションからデータベースに接続したり、EJB にアクセスしたりするための機能も提供しています。

ビジネスプロセス定義

業務の流れ（ビジネスプロセス）を定義したものです。

負荷分散機

Web ブラウザなどからのリクエストを一元的に受け付けて、同等の機能を持つ複数のサーバに転送して各サーバの負荷を分散させるための装置です。

フレームワーク

ある機能やサービスを規定するための枠組みです。アプリケーションフレームワークを使用することで、J2EE アプリケーションを容易に作成できます。

(マ行)

メッセージフロー

複数のシステム間でのメッセージの送受信による処理の流れです。
Cosminexus サービスプラットフォームでは、多様なシステム上のサービスをメッセージフローによってつなぐことで、サービス統合を実現します。

(ラ行)

リソースアダプタ

J2EE Connector Architecture によって、J2EE サーバと EIS を接続するための接続機能です。

リソースマネージャ

リソースを管理する機能です。DBMS などが該当します。

リダイレクタ

Web サーバに登録し Web コンテナへの接続を可能にするプラグインコンポーネント（ライブラリ）です。Web サーバに登録したリダイレクタによって、リクエストを URL パターンまたはラウンドロビン方式で複数の Web コンテナに振り分けて処理できます。

索引

数字

2 フェーズコミットメント 57

A

Apache2.0 41

Application Server 161

Application Server Enterprise 24

Application Server Standard 23

Application Server の構成ソフトウェアの位置づけ (SOAP アプリケーションを実行する場合) 44

Application Server の構成ソフトウェアの位置づけ (Web クライアントまたは EJB クライアントからのリクエストを処理する場合) 42

Application Server の構成ソフトウェアの位置づけ (メッセージによる処理の非同期実行をする場合) 45

C

cFramework 21

CORBA クライアントアプリケーション 113

Cosminexus 1, 2, 161

Cosminexus Application Development Plug-in 36

Cosminexus cFramework 36

Cosminexus Component Container

36, 42, 44, 161

Cosminexus Component Container - Client 38

Cosminexus Component Container - Redirector 38

Cosminexus Component Library 38

Cosminexus Component Transaction Monitor 39, 43, 161

Cosminexus DABroker Library 39, 43

Cosminexus Developer's Kit for Java 39

Cosminexus Manager 43, 161

Cosminexus Operator Plug-in 39

Cosminexus Performance Tracer

40, 43, 161

Cosminexus Reliable Messaging

40, 45, 161

Cosminexus RM 88

Cosminexus Service Coordinator

40, 48, 161

Cosminexus Service Development Plug-in 40

Cosminexus Studio cFramework 40

Cosminexus TPBroker 40

Cosminexus Web Services - Security 40, 44

Cosminexus XML Processor 41

Cosminexus が対応する標準仕様 160

Cosminexus コネクタ 36

Cosminexus サービスプラットフォーム 161

Cosminexus サービスプラットフォームで接続できるデータベース 142

Cosminexus サービスプラットフォームのシステム構成 133

Cosminexus サービスプラットフォームを構成するマシンにインストールする製品 133

Cosminexus サービスプラットフォームを構成する三つの環境 131

Cosminexus で実現できる非同期通信 78

Cosminexus の主な機能 56, 58

Cosminexus の製品構成 19

CTM 43, 162

D

DB Connector 162

DB Connector for Cosminexus RM 88

DB アダプタ 162

Developer 162

Developer Professional 28

Developer Standard 27

Developer の構成ソフトウェアの位置づけ 47

DOM API 57, 59

E

Eclipse 162
 Eclipse プラグイン 162
 EIS 43, 162
 EJB 162
 EJB クライアント 43, 162
 EJB クライアントアプリケーション 162
 EJB コンテナ 43, 56, 162
 EJB を利用した SOAP アプリケーション 114
 EJB を利用した SOAP アプリケーションの開発 57
 Enterprise Bean 163
 Enterprise Bean の優先制御, 流量制御および負荷分散 69
 Enterprise JavaBeans 162
 ERP パッケージとの連携 128
 ESB 124

H

HCSC-Manager Plug-in 39
 HCSC サーバ 130
 HiRDB Embedded Server Version 8 41
 Hitachi Web Server 41, 42

I

iWay Adapter 128

J

J2EE 37, 163
 J2EE 1.4 1, 66
 J2EE アプリケーション 6, 163
 J2EE アプリケーション実行環境 3
 J2EE アプリケーションの実行環境の種類 79
 J2EE アプリケーションのリロード 58
 J2EE サーバ 5, 163
 J2EE サーバから接続できるデータベース 140
 J2EE サーバ間のセッション情報の引き継ぎ 57

Java 163
 Java 2 Platform Standard Edition 5.0 39
 JavaServer Pages 163
 JavaVM 163
 Java 言語 56, 58
 JAXP 113
 JMS インタフェース 117
 JMS メッセージ 117
 JNDI によるルックアップ 57, 59
 JP1 144, 163
 JP1 との連携 58, 60, 144
 JSP 163
 JSP 事前コンパイル 58

M

Management Server 62, 163
 Message-driven Bean 163
 MQ アダプタ 128
 MQ プロトコルで接続するシステム 128
 MVC 163
 MyEclipse 111, 164
 MyEclipse for Cosminexus 41
 MyEclipse を使用したアプリケーションの開発 57, 59

N

N:1 リカバリシステム構成 58

O

Object Access アダプタ 127
 Object Wrapper システム 127
 OLTP 技術 68
 OLTP 技術の適用 57
 OpenTP1 システム 127

R

RMI-IIOP 164
 RPC を利用した SOAP アプリケーション 114
 RPC を利用した SOAP アプリケーションの開発 57

S

SAX API 57, 59
 Server Plug-in 37, 39, 111
 Service Architect 30
 Service Architect の構成ソフトウェアの位置づけ 49
 Service Platform 28
 Service Platform の構成ソフトウェアの位置づけ 48
 Session Bean 164
 Smart Composer 機能 62, 82, 164
 SOA 11, 124, 164
 SOAP 85, 164
 SOAP アプリケーション 77, 85, 164
 SOAP アプリケーションのクライアントとサーバに使用できる製品 86, 87
 SOAP アプリケーションの実行環境の例 85
 SOAP エンジン 85
 SOAP サービス 85
 SOAP メッセージ 85
 SOAP メッセージの完全性および秘匿性の保証 57
 SOAP メッセージの認証 57

T

TP1 アダプタ 127

U

uCosminexus Client 30, 164
 uCosminexus Operator 32, 164
 uCosminexus Service Adapter for Object Access 127
 uCosminexus Service Adapter for TP1 127
 uCosminexus TP1 Gateway 128
 uCosminexus Web Redirector 31, 165
 UDDI クライアントライブラリが提供する JAXR API 57
 UDDI レジストリ 115
 URL グループ 67

W

Web アプリケーション 165
 Web アプリケーションの流量制御 67
 Web 環境 52
 Web クライアント 42
 Web コンテナ 42, 56, 165
 Web サーバ 42, 52, 56, 165
 Web サービス 57, 77, 165
 Web サービスおよび SOAP 間通信 57
 Web サービスセキュリティ 86
 Web サービスセキュリティの技術を利用したシステムの例 87
 Web サービスに対応したアプリケーション 85
 Web サービス利用環境 77
 Web ブラウザ 52
 WS-Reliability 89, 165
 WS-Security 40
 WSDL 138

X

XML 暗号 116
 XML 署名データ 116
 XML 署名データの暗号化および復号化 57
 XML 署名データの生成および検証 57
 XML プロセッサ 57, 59
 XSLT トランスフォーマ 57, 59

あ

アーカイブファイルの作成 120
 アダプタ 126, 130
 アプリケーション / リソースの実行時属性の設定 122
 アプリケーション入れ替え 76
 アプリケーション開発 57, 59
 アプリケーション開発環境 3
 アプリケーションサーバ 6, 165
 アプリケーション実行環境 3
 アプリケーションの開発 61
 アプリケーションの設計 119
 アプリケーションの設定 58
 アプリケーションのデプロイ 58

い

イベント発行 58, 59

う

運用環境 131
運用管理機能 43
運用作業の自動化 76
運用設計 60

え

エンタープライズ・サービス・バス 124

お

オプション製品 22

か

開発環境 10, 131
開発環境のインスタントセットアップ機能
36, 112
開発環境の構成 108
開発環境の構成ごとにインストールする製品
109
開発環境のセットアップ 118
開発環境を構築する製品 21
稼働情報監視 57, 59
可用性 69
可用性向上 57, 59
簡易構築 58, 59
監査証跡情報 74
監査ログ 73
監査ログ出力 57, 59

き

業務アプリケーション 165
業務効率を向上させる運用管理の実現 75
業務指向スタイル 10
業務システムの一括運用 76
業務のコンポーネント化 70

く

クラスタソフトウェアとの連携 58, 59, 145

け

言語 52

こ

構成ソフトウェア 34
コールドスタンバイでの 1:1 の系切り替え
58, 59
コネクションシェアリング 57, 59
コネクションプーリング 57, 59
コンパイル 119

さ

サービス 11, 14, 21, 166
サービスインテグレーション 3, 124
サービスコンポーネント 130, 137, 166
サービスデプロイメント 3
サービス統合環境 3, 124
サービス統合環境を構築する製品 21
サービス閉塞 69
サービスユニット 82, 166
サービスリクエスト 29, 130, 137, 166
サービスリクエスト開発時に利用できる情報
138
サブレット 166

し

システム運用 58, 59
システム構成の一括定義 58, 59
システム構成の種類 80
システム構築 58, 59
システム設計 60
システム導入および拡張の容易化 71
システムの安定稼働 66
システムの運用と保守 62
システムの構築 62
システムの仕様（機能）検討 56
システムのライフサイクル 53, 55
実行 121

実行環境 5, 66, 131
 実行環境と開発環境の関係 15
 実行環境の移行性向上 58, 59
 実行環境を構築する製品 20
 自動アクション制御 58, 59
 障害発生時の可用性向上 70
 障害発生の未然防止 70
 信頼性の高い非同期通信 78

す

スケールアウト 80, 166
 スケールアウト構成 23
 スケールアウト構成の例 83
 スケールアウトシステム 80
 スケールアウトシステムの構成例 80
 スケールイン 80, 166
 スタブ 138

せ

性能解析トレース / 障害解析トレースの出力
 57, 59
 製品構成 19
 製品と構成ソフトウェアの対応 34
 セキュリティ管理 57, 59
 セッション情報の引き継ぎ 70
 セットアップウィザード 62, 82
 前提 OS 50

そ

相互スタンバイ構成 58, 59

た

耐障害性 69
 タイムアウトの設定 70
 対話型アプリケーション 14

て

データベース 52
 データベース監査証跡連携 57, 59
 データベースとの連携 140
 テスト 121

テスト用実行環境の開始 121
 デバッグ 119, 121
 デプロイ 166
 展開ディレクトリ形式のアプリケーション
 120

と

統合開発環境 10
 統合ユーザ管理 57, 76
 同時実行スレッド数制御 57
 ドメイン一括管理 58, 59
 トラブルシューティング 62
 トレース情報による性能解析 76

は

バックエンド 25, 81
 バッチアプリケーション 167
 バッチアプリケーション実行環境 3
 バッチサーバ 9, 59, 96, 167

ひ

ビジネスプロセス定義 167
 非同期通信アプリケーション 88

ふ

負荷分散 66
 負荷分散機 167
 複数実行環境の切り替え (UNIX の場合)
 58, 59
 フレームワーク 167
 フレームワークとライブラリ 57
 プログラミング 119
 フロントエンド 25, 81
 フロントエンド / バックエンド構成の例
 (CTM を使用する場合) 84
 フロントエンド / バックエンドシステム
 25, 81
 フロントエンド / バックエンドシステムの構
 成例 81
 フロントシステム 14
 分散トランザクションの実現 57

ほ

ほかの製品との連携 139

め

メッセージ 88
メッセージ送信先のキュー情報 138
メッセージの転送 89
メッセージフロー 124, 167
メッセージフローによるサービス統合 124
メッセージを利用した非同期通信アプリケーションの実行環境の例 88
メッセージを利用した非同期通信アプリケーションの送信側と受信側に使用できる製品 89
メッセージングを利用した SOAP アプリケーション 114
メッセージングを利用した SOAP アプリケーションの開発 57

ゆ

優先制御 66

り

リソースアダプタ 167
リソースアダプタのデプロイ 58, 59
リソース枯渇監視 57, 59
リソース接続とトランザクション管理 57, 59
リソースの設定 58, 59
リソースマネージャ 167
リダイレクタ 42, 167
リモート環境からの各種サーバの設定 58, 59
流量制御 66

れ

レガシーシステムとの連携 126

ろ

ローカルトランザクション 57, 59

ローカルマシンでのデバッグおよびリモートマシンでのデバッグ 57, 59
ログ/トレース収集 58, 59
ログの運用 76

ソフトウェアマニュアルのサービス ご案内

1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しています。

URL <http://www.hitachi.co.jp/soft/manual/>

ホームページのメニューは次のとおりです。

マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル	日立ソフトウェアマニュアルと製品群別CD-ROMマニュアルの仕様について記載しています。
マニュアルのご購入	マニュアルご購入時のお申し込み方法を記載しています。
オンラインマニュアル	一部製品のマニュアルをインターネットで公開しています。
サポートサービス	ソフトウェアサポートサービスお客様向けページでのマニュアル公開サービスを記載しています。
ご意見・お問い合わせ	マニュアルに関するご意見、ご要望をお寄せください。

2. インターネットでのマニュアル公開

2種類のマニュアル公開サービスを実施しています。

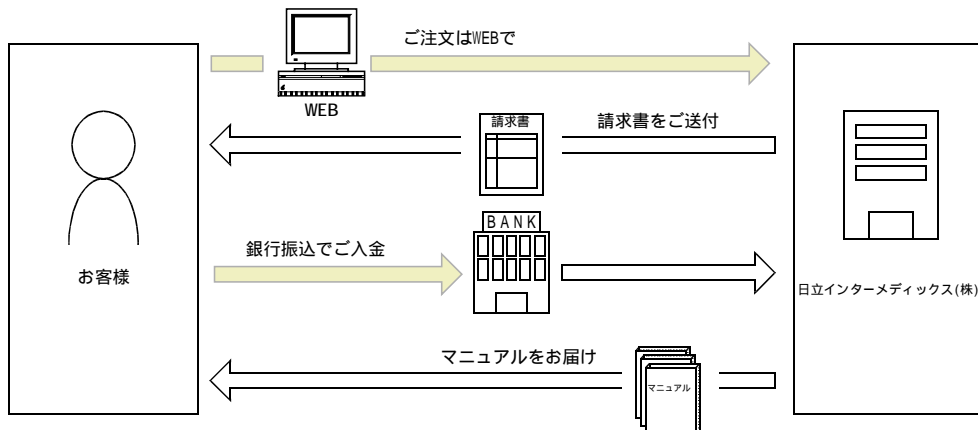
(1) マニュアル情報ホームページ「オンラインマニュアル」での公開

製品をよりご理解いただくためのご参考として、一部製品のマニュアルを公開しています。

(2) ソフトウェアサポートサービスお客様向けページでのマニュアル公開

ソフトウェアサポートサービスご契約のお客様向けにマニュアルを公開しています。公開しているマニュアルの一覧、本サービスの対象となる契約の種別などはマニュアル情報ホームページの「サポートサービス」をご参照ください。

3. マニュアルのご注文



マニュアル情報ホームページの「マニュアルのご購入」にアクセスし、お申し込み方法をご確認のうえWEBからご注文ください。ご注文先は日立インターメディアックス(株)となります。

ご注文いただいたマニュアルについて請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。

入金確認後7日以内にお届けします。在庫切れの場合は、納期を別途ご案内いたします。