



Sun Java™ System

Sun Java Enterprise System 2005T1 Présentation technique

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

Référence n° : 819-1925

Copyright © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs des brevets américains répertoriés à l'adresse <http://www.sun.com/patents> et un ou plusieurs des brevets supplémentaires ou des demandes de brevet en attente aux États-Unis et dans d'autres pays.

CE PRODUIT CONTIENT DES INFORMATIONS CONFIDENTIELLES ET DES SECRETS COMMERCIAUX DE SUN MICROSYSTEMS, INC. SON UTILISATION, SA DIVULGATION ET SA REPRODUCTION SONT INTERDITES SANS L'AUTORISATION EXPRESSE, ÉCRITE ET PRÉALABLE DE SUN MICROSYSTEMS, INC.

La distribution du logiciel peut s'accompagner de celle de composants mis au point par des tiers.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD sous contrat de licence par l'Université de Californie.

UNIX est une marque déposée aux États-Unis et dans d'autres pays, sous licence exclusive de X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, le logo Duke, le logo Java Coffee Cup, le logo Solaris, le logo SunTone Certified et le logo Sun™ ONE sont des marques commerciales ou déposées de Sun Microsystems, Inc. aux États Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées dans le cadre d'un contrat de licence et sont des marques commerciales ou déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

Legato, le logo Legato et Legato NetWorker sont des marques de fabrique ou des marques déposées de Legato Systems, Inc. Le logo Netscape Communications Corp est une marque de fabrique ou une marque déposée de Netscape Communications Corporation.

L'interface graphique utilisateur OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et détenteurs de licences. Sun reconnaît les efforts novateurs de Xerox en ce qui concerne la recherche et le développement du concept des interfaces visuelles ou graphiques dans le domaine informatique. Sun détient une licence non exclusive de Xerox sur l'interface utilisateur graphique de Xerox ; cette licence couvre également les détenteurs de licences Sun qui mettent en application l'interface graphique utilisateur OPEN LOOK et qui, en outre, se conforment aux contrats de licence écrits de Sun.

Les produits mentionnés dans ce manuel et les informations fournies sont soumis à la législation américaine en matière de contrôle des exportations et peuvent être soumis à la réglementation en vigueur dans d'autres pays dans le domaine des exportations et importations. L'utilisation d'armes nucléaires, de missiles, d'armes biologiques et chimiques ou d'armes nucléaires maritimes, qu'elle soit directe ou indirecte, est strictement interdite. Les exportations ou réexportations vers les pays sous embargo américain, ou vers des entités figurant sur les listes d'exclusion des exportations américaines, y compris, mais de manière non exhaustive, la liste des personnes refusées et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

CETTE DOCUMENTATION EST FOURNIE EN L'ÉTAT ET TOUTE CONDITION, DÉCLARATION ET GARANTIE EXPRESSE OU TACITE, Y COMPRIS TOUTE GARANTIE IMPLICITE DE QUALITÉ MARCHANDE, D'ADÉQUATION À UN BUT PARTICULIER OU D'ABSENCE DE CONTREFAÇON, SONT EXCLUES, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.

Sommaire

Liste des tableaux	7
Liste des figures	9
Préface	11
Public	12
Utilisation de la documentation	12
Conventions	14
Ressources sur Internet	14
Comment signaler les problèmes	15
Sun attend vos commentaires.	15
Chapitre 1 Introduction à Java Enterprise System	17
Nécessité de Java Enterprise System	18
Composants Java Enterprise System	20
Composants de service du système	21
Composants de qualité de service	23
Composants de disponibilité	23
Composants d'accès	24
Composants administratifs	25
Composants partagés	25
Utilisation de Java Enterprise System	26
Cycle de vie de la solution Java Enterprise System	26
Scénarios d'adoption de Java Enterprise System	29
Termes clés de ce chapitre	31

Chapitre 2 Architectures de la solution Java Enterprise System	33
Structure architecturale de Java Enterprise System	34
Dimension 1 : Dépendances des services d'infrastructure	35
Niveaux de services d'infrastructure	35
Composants des services d'infrastructure de Java Enterprise System	39
Dépendances des services d'infrastructure de Java Enterprise System	40
Dimension 2 : Niveaux logiques	41
Description des niveaux logiques	42
Indépendance physique et logique	43
Architecture à plusieurs niveaux appliquée aux composants système	44
Dimension 3 : Qualité de service	45
Qualités de service	46
Composants de qualité de service de Java Enterprise System	47
Logiciel Sun Cluster	48
Synthèse des trois dimensions architecturales	50
Exemple d'architecture de solution Java Enterprise System	51
Scénario de communication d'entreprise	51
Architecture logique de l'exemple de scénario	52
Architecture de déploiement de l'exemple de scénario	54
Termes clés de ce chapitre	54
Chapitre 3 Fonctions d'intégration de Java Enterprise System	57
Le programme d'installation intégré de Java Enterprise System	58
Contrôle de logiciels préexistants	59
Contrôle de dépendance	59
Configuration initiale	60
Désinstallation	60
Services intégrés d'identité et de sécurité	60
Identité unique	61
Authentification et connexion unique	62
Authentification	62
Connexion unique	63
Autorisation	64
Termes clés de ce chapitre	65
Chapitre 4 Tâches du cycle de vie de la solution Java Enterprise System	67
Déploiement préalable	69
Déploiement	70
Conception du déploiement	71
Architecture de déploiement	71
Spécifications d'implémentation	73
Plans d'implémentation	73

Mise en œuvre du déploiement	73
Configuration matérielle	74
Installation logicielle, mise à niveau et migration	74
Configuration système et personnalisation	74
Développement et intégration	75
Test des prototypes et des pilotes	75
Mise en production	76
Déploiement ultérieur	76
Termes clés de ce chapitre	77
Appendice A Liste de références : composants Java Enterprise System	79
Description des composants de service du système	80
Sun Java System Access Manager 6 2005T1	80
Sun Java System Application Server Enterprise Edition 8 2005T1	81
Sun Java System Calendar Server 6 2005T1	82
Sun Java System Directory Server 5 2005T1	82
Sun Java System Instant Messaging 7 2005T1	82
Sun Java System Message Queue 3 2005T1	83
Sun Java System Messaging Server 6 2005T1	84
Sun Java System Portal Server 6 2005T1	84
Sun Java System Web Server 6.1 2005T1	85
Description des composants de qualité de service	85
Description des composants de disponibilité	85
Sun Cluster 3.1 9/04 et Sun Cluster Agents	85
High Availability Session Store 2005T1	86
Description des composants d'accès	86
Sun Java System Communications Express 2005T1	87
Sun Java System Connector pour Microsoft Outlook 6 2005T1	87
Sun Java System Directory Proxy Server 5 2005T1	87
Sun Java System Portal Server Secure Remote Access 6 2005T1	88
Description des composants administratifs	88
Sun Java System Administration Server (et Console) 5 2005T1	89
Sun Java System Directory Preparation Tool 2005T1	89
Sun Java System Delegated Administrator 6 2005T1	89
Sun Remote Services Net Connect 2005T1	89
Composants partagés	90
Index	93

Liste des tableaux

Tableau 1	Documentation Java Enterprise System	13
Tableau 2	Conventions de style	14
Tableau 1-1	Composants de service du système Java ES	22
Tableau 1-2	Composants de disponibilité Java ES	24
Tableau 1-3	Composants d'accès Java ES	24
Tableau 1-4	Composants administratifs Java ES	25
Tableau 1-5	Catégories d'utilisateurs Java ES pour les tâches de cycle de vie	28
Tableau 1-6	Problèmes liés aux différents scénarios d'adoption Java ES	30
Tableau 2-1	Relations entre les composants de service du système Java ES	40
Tableau 2-2	Qualités de service affectant l'architecture de la solution	46
Tableau 2-3	Composants de qualité de service et qualités système influencées	48
Tableau 2-4	Résumé des exigences d'entreprise : scénario de communication	51

Liste des figures

Figure 1-1	Prise en charge requise pour les applications d'entreprise distribuées	19
Figure 1-2	Catégories de composants Java Enterprise System	21
Figure 1-3	Phases du cycle de vie d'une solution et catégories d'utilisateurs	27
Figure 2-1	Dimensions d'une architecture de solution Java ES	35
Figure 2-2	Dimension 1 : Niveaux de services d'infrastructure	36
Figure 2-3	Composants de service du système Java ES	39
Figure 2-4	Dimension 2 : Niveaux logiques pour les applications d'entreprise distribuées	41
Figure 2-5	Messaging Server: Exemple d'architecture à plusieurs niveaux	45
Figure 2-6	Conception de disponibilité à l'aide de nœuds Sun Cluster	49
Figure 2-7	Architecture logique d'un scénario de communication d'entreprise	53
Figure 3-1	Une entrée utilisateur unique prend en charge de nombreux services	61
Figure 3-2	Séquence d'authentification	63
Figure 3-3	Séquence d'autorisation	64
Figure 4-1	Tâches du cycle de vie d'une solution	68
Figure 4-2	Spécification d'un scénario de déploiement	70
Figure 4-3	Conversion d'un scénario de déploiement en une architecture de déploiement	71

Préface

Le manuel Présentation technique de Sun Sun Java™ Enterprise System décrit les principes techniques et conceptuels de Java Enterprise System. Il présente également les composants de Java Enterprise System, son architecture, ses processus et ses fonctions.

L'objectif de cette présentation consiste à clarifier les concepts et la terminologie techniques employés dans la documentation de Java Enterprise System. Les principaux termes techniques sont expliqués dans la section « Termes clés » de chaque chapitre, qui décrit comment ces termes sont utilisés dans le contexte Java Enterprise System.

La préface comprend les sections suivantes :

- « Public », page 12
- « Utilisation de la documentation », page 12
- « Conventions », page 14
- « Ressources sur Internet », page 14
- « Comment signaler les problèmes », page 15
- « Sun attend vos commentaires. », page 15

Public

Le manuel *Présentation technique de Sun Java Enterprise System 2005T1* a été rédigé à l'attention des personnes chargées de la conception, du déploiement ou de la maintenance des solutions logicielles basées sur Java Enterprise System. Cela représente un vaste public comprenant des analystes d'exploitation, des architectes système, des ingénieurs de projet et des administrateurs système.

Les personnes lisant ce manuel doivent avoir quelques connaissances sur les technologies suivantes :

- Concepts généraux de gestion de réseaux
- Notions élémentaires de sécurité relatives à l'authentification et à l'autorisation
- Le langage Java, les composants Java 2 Standard Edition et les composants Java 2 Enterprise Edition

Utilisation de la documentation

Les manuels relatifs à Java Enterprise System sont disponibles en ligne sous la forme de fichiers PDF (Portable Document Format) et HTML (HyperText Markup Language). Ces deux formats sont compatibles avec les aides technologiques proposées aux utilisateurs souffrant d'un handicap physique. La documentation Sun™ est disponible sur le site Web à l'adresse suivante :

<http://docs.sun.com>.

Pour accéder à la documentation Java Enterprise System, saisissez l'URL suivant :

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

Le tableau ci-dessous répertorie les manuels du système figurant dans la documentation relative à Java Enterprise System. La colonne de gauche indique le nom et le numéro de chaque document, et la colonne de droite décrit le contenu général du document.

Tableau 1 Documentation Java Enterprise System

Document	Contenu
<i>Sun Java Enterprise System 2005T1 – Notes de version</i> http://docs.sun.com/doc/819-0057	Ce manuel contient les informations les plus récentes sur Java Enterprise System, y compris les problèmes connus. De plus, chaque composant dispose de notes de version qui lui sont propres.
<i>Présentation de la documentation de Sun Java Enterprise System 2005T1</i> http://docs.sun.com/doc/819-0055	Ce guide décrit la documentation relative à Java Enterprise System. Il comporte des liens renvoyant à la documentation associée aux composants.
<i>Présentation technique de Sun Java Enterprise System 2005T1</i> http://docs.sun.com/doc/819-0061	Ce guide décrit les principes techniques et conceptuels de Java Enterprise System. Il décrit les composants, l'architecture, les processus et les fonctions.
<i>Guide de planification du déploiement de Sun Java Enterprise System 2005T1</i> http://docs.sun.com/doc/819-0058	Ce manuel indique brièvement les modalités de planification et de conception des solutions de déploiement d'entreprise basées sur Java Enterprise System. Il présente quelques principes et concepts fondamentaux de la planification et de la conception de déploiement, traite du cycle de vie d'une solution et fournit des exemples et des stratégies de haut niveau à utiliser lors de la planification de solutions basées sur Java Enterprise System.
<i>Guide de gestion des utilisateurs de Sun Java Enterprise System 2005T1</i> http://docs.sun.com/doc/817-5761	Ce document vous aide à planifier, déployer et gérer les informations relatives aux utilisateurs de votre solution Java Enterprise System. Il complète le manuel <i>Guide de planification du déploiement de Sun Java Enterprise System 2005T1</i> en décrivant les problèmes de gestion utilisateur dans chaque phase du cycle de vie d'une solution.
<i>Sun Java Enterprise System 2003T4 - Série d'exemples de déploiement : Scénario d'évaluation</i> http://docs.sun.com/doc/819-0059	Ce manuel décrit les procédures d'installation de Java Enterprise System sur un ordinateur, de mise en place des principaux services réseau partagés et de configuration des comptes utilisateur pouvant accéder à ces services.
<i>Guide d'installation de Sun Java Enterprise System 2005T1</i> http://docs.sun.com/doc/819-0056	Ce manuel vous guide tout au long de la procédure d'installation de Java Enterprise System pour le système d'exploitation Solaris™ ou Linux. Il vous explique comment sélectionner les composants à installer, les configurer et comment vous assurer de leur fonctionnement correct.
<i>Guide de migration et de mise à niveau de Sun Java Enterprise System 2005T1</i> http://docs.sun.com/doc/819-0062	Ce manuel fournit des informations et des instructions sur la mise à niveau de Java Enterprise System pour l'environnement d'exploitation Solaris™ ou Linux.
<i>Glossaire de Sun Java Enterprise System</i> http://docs.sun.com/doc/816-6873	Ce document définit les termes utilisés dans la documentation de Java Enterprise System.

Outre les documents système cités dans ce tableau, la documentation de Java Enterprise System inclut une documentation propre à chaque composant pour la plupart des composants de Java Enterprise System. Reportez-vous au document *Présentation de la documentation de Sun Java Enterprise System 2005T1* pour de plus amples détails.

Conventions

Les conventions de style employées dans le présent manuel sont décrites dans le tableau ci-dessous.

Tableau 2 Conventions de style

Style	Signification	Exemples
AaBbCc123 (largeur fixe)	Interfaces API et éléments linguistiques, balises HTML, localisateurs URL de sites Web, noms de commande, noms de fichier, chemins d'accès aux répertoires, éléments affichés sur l'écran de l'ordinateur, exemple de code.	Modifiez votre fichier <code>.login</code> . Pour établir la liste de tous les fichiers, entrez <code>ls -a</code> . % Vous avez du courrier.
<i>AaBbCc123</i> (italique)	Titres d'ouvrage. Mots ou termes nouveaux. Mots importants. Variables de ligne de commande à remplacer par des valeurs ou noms réels.	Lisez le chapitre 6 du <i>guide de l'utilisateur</i> . Il s'agit d'options de <i>classe</i> . Vous <i>devez</i> avoir le statut de superutilisateur pour pouvoir effectuer cette opération. Le fichier est stocké dans le répertoire <i>install-dir/bin</i> .

Ressources sur Internet

Des informations concernant le logiciel Java Enterprise System ainsi que ses composants sont disponibles sur le site Web suivant :

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

Comment signaler les problèmes

Si vous rencontrez des problèmes avec Java Enterprise System, contactez le service clientèle de Sun de l'une des manières suivantes :

- En faisant appel aux services d'assistance logicielle Sun en ligne à l'adresse suivante :

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

Ce site contient des liens vers la base de connaissances, le centre d'assistance en ligne et ProductTracker, ainsi que vers des programmes de maintenance et des coordonnées pour l'assistance.

- En composant le numéro de téléphone indiqué sur votre contrat de maintenance.

Afin de vous aider au mieux à résoudre votre problème, nous vous suggérons de réunir les informations suivantes lorsque vous contactez l'assistance technique de Sun :

- la description du problème, en particulier les situations dans lesquelles il se produit et son impact sur vos opérations ;
- le type de machine, les versions du système d'exploitation et du produit, y compris les patches et autres logiciels pouvant avoir un lien avec le problème ;
- la procédure détaillée des méthodes utilisées pour reproduire le problème ;
- tous les journaux d'erreur ou vidages de la mémoire.

Sun attend vos commentaires.

Afin d'améliorer sa documentation, Sun vous encourage à faire des commentaires et à apporter des suggestions.

Pour nous faire part de vos commentaires, rendez-vous sur le site <http://docs.sun.com>, puis cliquez sur Envoyer les commentaires.

Dans le formulaire en ligne, indiquez le titre et le numéro du document.

Ce numéro est constitué de sept ou neuf chiffres et figure sur la page de titre du manuel ou en haut du document. Par exemple, le titre de ce manuel est *Sun Présentation technique de Sun Java Enterprise System 2005T1*, et son numéro de référence est 819-1925.

Sun attend vos commentaires.

Introduction à Java Enterprise System

Sun Java™ Enterprise System (Java ES) est un ensemble de composants logiciels qui fournissent les services nécessaires à la prise en charge des applications d'entreprise distribuées dans un environnement réseau ou Internet. Dans ce manuel, ces applications sont qualifiées d'applications d'entreprise distribuées.

Java Enterprise System est également un logiciel Sun spécifique qui procure une méthode de livraison et une stratégie d'entreprise et de tarification. Toutefois, ce manuel traite essentiellement des composants logiciels de Java Enterprise System et des services qu'ils fournissent.

Ce chapitre présente Java Enterprise System et les tâches liées à l'utilisation du système. Il contient les sections suivantes :

- « Nécessité de Java Enterprise System », page 18
- « Composants Java Enterprise System », page 20
- « Utilisation de Java Enterprise System », page 26
- « Termes clés de ce chapitre », page 31

Nécessité de Java Enterprise System

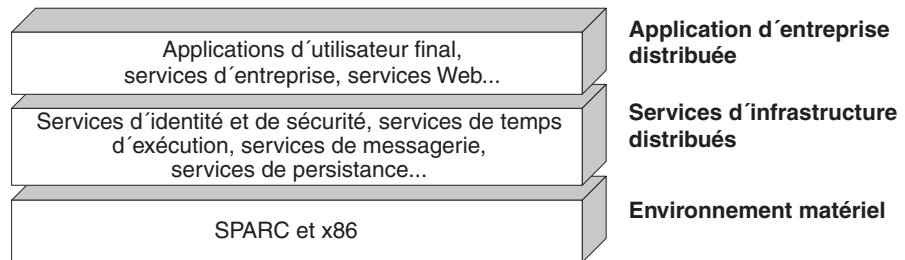
Les besoins actuels d'une entreprise exigent des solutions logicielles qui sont distribuées sur un environnement réseau ou Internet et qui possèdent des niveaux élevés de performance, de disponibilité, de sécurité, d'évolutivité et d'entretien.

Java Enterprise System fournit les services d'infrastructure requis pour la prise en charge de ces *applications d'entreprise distribuées*, c'est-à-dire des applications qui possèdent généralement les caractéristiques suivantes :

- **Distribuée** : l'application est constituée de *composants* logiciels en interaction qui sont déployés sur un environnement réseau pouvant inclure des sites géographiquement distants. Ces composants distribués, qui sont exécutés sur divers ordinateurs de l'environnement, fonctionnent ensemble afin de procurer des fonctions d'entreprise propres aux *utilisateurs finals* et aux autres applications d'entreprise.
- **Niveau entreprise** : l'étendue et l'échelle de l'application satisfont les besoins d'un environnement de production ou d'un fournisseur d'accès à Internet. Généralement, l'application est déployée dans toute l'entreprise, en intégrant la plupart des services, des opérations et des processus dans un système logiciel unique. Elle doit satisfaire des exigences élevées en termes de qualité de service concernant les performances, la disponibilité, la sécurité, l'évolutivité et l'entretien.

Les applications d'entreprise distribuées requièrent un ensemble sous-jacent de *services* d'infrastructure qui permettent à leurs composants distribués de communiquer entre eux, de coordonner leur travail, d'implémenter un accès sécurisé, etc. Ces services d'infrastructure sont alors pris en charge par un environnement matériel d'ordinateurs et de liaisons réseau. Cet environnement matériel comprend les architectures matérielles SPARC® et x86 (Intel et AMD).

La structure en couches global est illustrée à la figure suivante. Pour la plus grande partie, Java Enterprise System fournit la couche de services d'infrastructure distribués illustrés à la [Figure 1-1](#). Cependant, les services Java Enterprise System comportent également plusieurs services de niveau application qui sont accessibles aux utilisateurs finals.

Figure 1-1 Prise en charge requise pour les applications d'entreprise distribuées

Parmi les services fournis par Java Enterprise System, il faut citer les services suivants :

- **Services de portail** : ces services permettent aux employés, aux télétravailleurs, aux travailleurs du savoir, aux partenaires commerciaux, aux fournisseurs et aux clients d'accéder aux ressources d'entreprise depuis n'importe quel emplacement intérieur ou extérieur au réseau d'entreprise. Ces services procurent des fonctionnalités d'accès aux communautés d'utilisateurs, quels que soient l'heure et l'endroit, en assurant une intégration, une agrégation, une sécurité, un accès mobile et une recherche personnalisés.
- **Services de communication et de collaboration** : ces services permettent un échange d'informations sécurisé entre les diverses communautés d'utilisateurs. Parmi les fonctionnalités spécifiques, il faut citer la messagerie, la collaboration en temps réel, telle que la messagerie instantanée et la tenue de conférence, ainsi que la planification dans le contexte de l'environnement de travail de l'utilisateur.
- **Services de sécurité et d'identité réseau** : ces services améliorent la sécurité et la protection des ressources d'informations clés de l'entreprise en garantissant l'application de stratégies de contrôle d'accès appropriées dans toutes les communautés; toutes les applications et tous les services à un niveau général. Ces services fonctionnent avec un référentiel permettant de stocker et de gérer les profils d'identité, les droits d'accès ainsi que les informations sur les applications et les ressources réseau.
- **Services de conteneur Web et services applicatifs** : ces services permettent aux composants distribués de communiquer entre eux pendant leur exécution et de prendre en charge le développement, le déploiement et la gestion des applications pour une large palette de serveurs, de clients et de périphériques. Ces services sont basés sur la plate-forme Java 2, Enterprise Edition (J2EE™).

Java Enterprise System fournit également des services qui améliorent la disponibilité, l'évolutivité, l'entretien et d'autres qualités des applications ou du système. Parmi les fonctions de qualité de service fournies par Java Enterprise System, il faut citer les suivantes :

- **Services de disponibilité** : ces services assurent une disponibilité quasi continue pour les composants d'applications et les composants d'infrastructure qui les prennent en charge.
- **Services d'accès** : ces services fournissent un accès par Internet ou par un navigateur aux services Java Enterprise System.
- **Services administratifs** : ces services permettent d'entretenir et de régler les performances des applications prises en charge par Java Enterprise System.

Vous pouvez déployer un ou plusieurs services Java Enterprise System, chaque service pouvant inclure plusieurs composants Java Enterprise System.

Composants Java Enterprise System

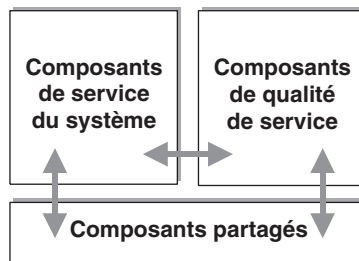
Java Enterprise System est en fait l'intégration de plusieurs logiciels précédemment distincts dans un seul et même système. Les composants de ce système ont été testés ensemble pour vérifier leur interopérabilité. Leur intégration est facilitée par un certain nombre de fonctions système :

- Tous les composants sont synchronisés sur un ensemble commun de bibliothèques partagées.
- Tous les composants Java Enterprise System sont installés au moyen d'un seul programme d'installation.
- Tous les composants Java Enterprise System peuvent partager un système intégré de gestion de sécurité et d'identité utilisateur.

Ces fonctions sont expliquées dans les chapitres suivants de ce manuel. Cette section a pour objet de présenter les divers composants intégrés à Java Enterprise System. Ces *composants système* peuvent être regroupés en trois catégories principales, comme illustré à la figure ci-après :

- **Composants de service du système** : ces composants fournissent les services d'infrastructure principaux de Java Enterprise System qui prennent en charge les applications d'entreprise distribuées.
- **Composants de qualité de service** : ces composants améliorent la disponibilité, la sécurité, l'évolutivité, l'entretien et les autres qualités des composants de service du système et des composants d'applications distribuées.
- **Composants partagés** : ces composants procurent l'environnement dans lequel la plupart des composants de service du système et de qualité de service sont exécutés.

Figure 1-2 Catégories de composants Java Enterprise System



Composants de service du système

Plusieurs composants Java Enterprise System fournissent les principaux services qui prennent en charge les solutions logicielles distribuées. Ces *services système* incluent les services de portail, les services de communication et de collaboration, les services d'identité et de sécurité, les services de conteneur Web et les services applicatifs J2EE.

Le tableau suivant présente une brève description des *composants de service du système* et des services distribués qu'ils fournissent. Chaque composant de service du système est un processus serveur à unités d'exécution multiples pouvant prendre en charge de nombreux clients. Pour en savoir plus sur chaque composant, reportez-vous à la section « [Description des composants de service du système](#) », page 80.

Tableau 1-1 Composants de service du système Java ES

Composant	Services système fournis
Sun Java System Access Manager	Fournit des services de gestion d'accès et d'administration des identités numériques. Les services de gestion d'accès englobent l'authentification (y compris la connexion unique) et l'autorisation, basée sur des rôles, d'accéder aux applications ou aux services. Les services d'administration regroupent l'administration centralisée des comptes utilisateur individuels, des rôles, des groupes et des stratégies.
Sun Java System Application Server	Fournit des services de conteneur J2EE pour les composants Enterprise JavaBeans™ (EJB), tels que les beans session, les beans entité et les beans gérés par message. Le conteneur fournit les services d'infrastructure requis pour assurer l'interaction des composants distribués à configuration groupée, faisant d'Application Server une plate-forme de développement et d'exécution des services Web et des applications d'e-commerce. Application Server fournit également des services de conteneur Web.
Sun Java System Calendar Server	Fournit des services de calendrier et de planification aux utilisateurs finals et aux groupes d'utilisateurs finals. Calendar Server comporte un client, basé sur le navigateur, qui interagit avec le serveur.
Sun Java System Directory Server	Fournit un référentiel central permettant de stocker et de gérer les informations du réseau intranet et d'Internet, telles que les profils d'identité (employés, clients, fournisseurs, etc.), les justificatifs d'identité des utilisateurs (certificats de clé publique, mots de passe, codes confidentiels), les droits d'accès, les informations sur les ressources des applications et les informations sur les ressources réseau.
Sun Java System Instant Messaging	Assure une communication sécurisée en temps réel entre les utilisateurs finals par le biais de la messagerie instantanée (discussion), de la tenue de conférences, des alertes, des forums, des sondages et du transfert de fichiers. Ce service comporte un gestionnaire de présence qui indique aux utilisateurs qui est actuellement en ligne ; il comporte un client basé sur le navigateur qui interagit avec le serveur.
Sun Java System Message Queue	Assure une messagerie fiable et asynchrone entre les applications et les composants distribués à configuration dispersée. Message Queue implémente la spécification Java Message Service (JMS) API et ajoute des fonctions d'entreprise, telles que la sécurité, l'évolutivité et l'administration à distance.
Sun Java System Messaging Server	Fournit une messagerie de stockage et de transfert sécurisée et fiable de haute capacité qui prend en charge les applications vocales, vidéo, de messagerie, de fax et de pager. Messaging Server peut accéder simultanément à plusieurs emplacements de stockage des messages et il fournit un filtrage de contenu permettant de rejeter les e-mails non sollicités et d'empêcher les attaques de virus.

Tableau 1-1 Composants de service du système Java ES (*suite*)

Composant	Services système fournis
Sun Java System Portal Server	Fournit des services de portail clés, tels que la personnalisation et l'agrégation de contenu, aux clients de navigateur qui accèdent aux services ou aux applications d'entreprise. Portal Server fournit également un moteur de recherche configurable.
Sun Java System Web Server	Fournit des services de conteneur Web J2EE™ pour les composants Web Java tels que les composants Java Servlet et JavaServer Pages™ (JSP™). Web Server prend également en charge d'autres technologies d'applications Web fournissant un contenu Web statique et dynamique telles que les scripts CGI et les pages ASP Sun Java System.

Composants de qualité de service

Outre les composants de service du système présentés dans le [Tableau 1-1](#), Java Enterprise System comporte plusieurs composants destinés à améliorer la qualité des services fournis par les composants de service du système ou par les composants d'applications personnalisées. Ces *composants de qualité de service* sont regroupés dans les catégories suivantes :

- composants de disponibilité ;
- composants d'accès ;
- composants administratifs.

Composants de disponibilité

Les composants de disponibilité assurent un temps de disponibilité quasi continu des composants de service du système et des composants d'applications. Le tableau suivant présente les composants de disponibilité inclus dans Java Enterprise System ainsi que les services qu'ils fournissent. Pour en savoir plus sur chaque composant, reportez-vous à la section « [Description des composants de disponibilité](#) », page 85.

Tableau 1-2 Composants de disponibilité Java ES

Composant	Services de disponibilité fournis
Sun Cluster	Fournit des services de disponibilité et d'évolutivité élevées pour Java Enterprise System, les applications exécutées en haut de l'infrastructure Java Enterprise System et l'environnement matériel dans lequel ces deux entités sont déployées.
High Availability Session Store	Fournit un magasin de données qui rend disponibles les données d'applications, en particulier les données de l'état de session, même en cas de panne.

Composants d'accès

Les composants d'accès assurent un accès frontal aux services système (fréquemment un accès sécurisé depuis des sites Internet situés hors du pare-feu de l'entreprise). Outre le fait de fournir ce type d'accès, la plupart de ces composants comportent également une fonction de routage. Le tableau suivant présente les composants d'accès inclus dans Java Enterprise System ainsi que les services qu'ils fournissent. Pour en savoir plus sur chaque composant, reportez-vous à la section « [Description des composants d'accès](#) », page 86.

Tableau 1-3 Composants d'accès Java ES

Composant	Services d'accès fournis
Sun Java System Directory Proxy Server	Fournit des services de sécurité pour Directory Server depuis l'extérieur d'un pare-feu d'entreprise. Directory Proxy Server assure un contrôle amélioré de l'accès aux répertoires, une compatibilité de schéma, un filtrage des attributs et un routage pour plusieurs instances de Directory Server.
Sun Java System Portal Server Secure Remote Access	Fournit un accès Internet sécurisé depuis l'extérieur d'un pare-feu d'entreprise au contenu et services de Portal Server, y compris les portails internes et les applications Internet.
Sun Java System Communications Express	Fournit un accès Web à Messaging Server, Calendar Server et Directory Server selon la configuration.
Sun Java System Connector pour Microsoft Outlook	Fournit aux clients de bureau utilisant Microsoft Outlook une interface pour Messaging Server et Calendar Server.

Composants administratifs

Les composants administratifs assurent des fonctions de gestion, telles que la configuration et le contrôle, pour les services système. Le tableau suivant présente les composants administratifs inclus dans Java Enterprise System ainsi que les services qu'ils fournissent. Pour en savoir plus sur chaque composant, reportez-vous à la section « [Description des composants administratifs](#) », page 88.

Tableau 1-4 Composants administratifs Java ES

Composant	Services administratifs fournis
Sun Java System Administration Server (et Console)	Fournit un outil d'administration graphique qui permet de configurer et de gérer Directory Server et Messaging Server.
Sun Java System Directory Preparation Tool	Fournit un script permettant de configurer Directory Server avec le schéma nécessaire à la création et à la configuration des utilisateurs pour Messaging Server et Calendar Server.
Sun Java System Delegated Administrator	Fournit des outils d'interface graphique et de ligne de commande permettant de compléter les entrées utilisateur dans Directory Server avec les attributs utilisateur requis par Messaging Server et Calendar Server.
Sun Remote Services Net Connect	Procure une fonctionnalité de contrôle à distance.

Composants partagés

Java Enterprise System comporte plusieurs bibliothèques partagées installées localement dont dépendent de nombreux composants de service du système et de nombreux composants de qualité de service. Les *composants partagés* Java Enterprise System fournissent des services locaux aux composants Java Enterprise System fonctionnant sur le même ordinateur hôte.

Les composants partagés sont fréquemment utilisés pour assurer la portabilité sur différents systèmes d'exploitation. Voici quelques exemples de composants partagés Java Enterprise System : plate-forme Java 2, Standard Edition (plate-forme J2SE™), Netscape Portable Runtime (NSPR), Network Security Services (NSS), Network Security Services pour Java (JSS), etc. Pour obtenir la liste complète, reportez-vous à la section « [Composants partagés](#) », page 90.

Utilisation de Java Enterprise System

La création de solutions d'entreprise basées sur le logiciel Java Enterprise System implique plusieurs tâches standard. L'étendue et la difficulté de ces tâches varient selon le point de départ choisi pour l'adoption de Java Enterprise System et la nature de la solution que vous essayez de créer et de déployer.

Cette section aborde deux aspects de l'utilisation de Java Enterprise System : le cycle de vie de la solution Java Enterprise System et les divers scénarios d'adoption qui sont généralement impliqués.

Cycle de vie de la solution Java Enterprise System

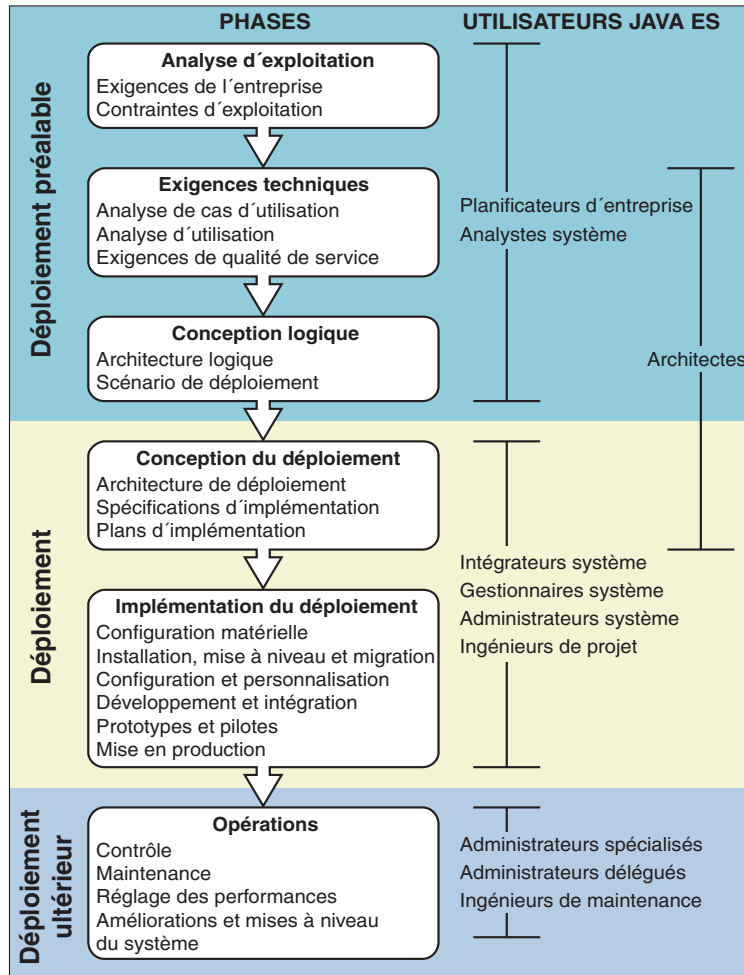
Les tâches impliquées dans la création de solutions d'entreprise basées sur le logiciel Java Enterprise System peuvent être divisées en plusieurs phases, comme illustré à la [Figure 1-3, page 27](#). Cette figure présente également la catégorie d'utilisateurs Java Enterprise System qui effectue les diverses tâches en règle générale.

Les phases du cycle de vie illustrées à la [Figure 1-3](#) peuvent être classées dans les groupes généraux suivants :

- **Déploiement préalable** : au cours de cette phase, un besoin d'entreprise est converti en un scénario de déploiement : une architecture logique et un ensemble d'exigences en termes de qualité de service. Le scénario de déploiement fait office de spécification pour la conception d'une architecture de déploiement.
- **Déploiement** : au cours de ces phases, un scénario de déploiement est converti en une architecture de déploiement. Cette architecture peut servir de base pour l'approbation et la budgétisation d'un projet. L'architecture de déploiement constitue également la base d'une spécification d'implémentation qui fournit les détails requis pour déployer (créer, tester et démarrer) une solution logicielle dans un environnement de production.
- **Déploiement ultérieur** : au cours de la phase d'opérations, une solution déployée est exécutée dans les conditions de production, puis contrôlée et optimisée en termes de performances. La solution déployée est également mise à niveau pour inclure une nouvelle fonctionnalité si nécessaire.

Le cycle de vie de la solution et les tâches de chacune des phases illustrés à la [Figure 1-3](#) sont expliqués de manière plus détaillée au [chapitre 4, Tâches du cycle de vie de la solution Java Enterprise System](#).

Figure 1-3 Phases du cycle de vie d'une solution et catégories d'utilisateurs



La [Figure 1-3](#) présente les utilisateurs Java Enterprise System qui effectuent généralement les tâches illustrées pour les phases du cycle de vie. Si vous utilisez Java Enterprise System, votre fonction doit correspondre à une ou plusieurs catégories d'utilisateurs représentées à la [Figure 1-3](#). Le tableau suivant décrit les connaissances et l'expérience correspondant à chaque catégorie d'utilisateurs.

Tableau 1-5 Catégories d'utilisateurs Java ES pour les tâches de cycle de vie

Utilisateur	Connaissances et expérience	Phases
Planificateur d'entreprise	Possède des connaissances générales et non des connaissances techniques approfondies.	Analyse d'exploitation
Analyste système	Sait en quoi consiste la direction stratégique d'une entreprise. Connaît les processus, les objectifs et les exigences d'entreprise.	Exigences techniques Conception logique
Architecte	Est très technique. Possède de vastes connaissances sur les architectures de déploiement. Connaît les technologies les plus récentes. Comprend les exigences et les contraintes d'entreprise.	Conception logique Conception du déploiement
Intégrateur système	Est très technique.	Conception du déploiement
Ingénieur de projet	Connaît très bien les environnements de technologie d'information.	Implémentation du déploiement
Administrateur système	A de l'expérience en matière d'implémentation de solutions logicielles distribuées.	
Gestionnaire système	Connaît les protocoles, les périphériques, la sécurité et l'architecture réseau. Connaît les langages de script et de programmation.	
Administrateur système spécialisé	Possède des connaissances sur le produit ou des connaissances techniques spécialisées.	Opérations
Administrateur délégué	Connaît le matériel, les plates-formes, les répertoires et les bases de données.	
Ingénieur de maintenance	Dispose de l'expérience requise en matière de contrôle, de dépannage et de mise à niveau des logiciels. Connaît l'administration système pour les plates-formes du système d'exploitation.	

Scénarios d'adoption de Java Enterprise System

Les besoins de l'entreprise qui mènent à l'adoption de Java Enterprise System peuvent varier considérablement. Toutefois, l'objectif de haut niveau de la quasi-totalité des déploiements Java Enterprise System correspond à l'un des *scénarios d'adoption* suivants :

- **Nouveau système** : en commençant avec un système logiciel inexistant, vous déployez le logiciel Java Enterprise System pour prendre en charge une nouvelle solution d'entreprise.
- **Amélioration** : en commençant avec une infrastructure de technologie d'information existante, vous remplacez une, plusieurs ou toutes les parties de ce système par le logiciel Java Enterprise System. En règle générale, vous remplacez les systèmes ou les sous-systèmes parce qu'ils sont trop complexes, trop limités ou trop coûteux à entretenir. Par exemple, vous aurez peut-être besoin d'une meilleure sécurité, d'une disponibilité plus élevée, d'une évolutivité accrue, de plus de flexibilité, d'une moindre complexité, de fonctionnalités supplémentaires (comme la connexion unique) ou d'une meilleure utilisation des ressources informatiques. En d'autres termes, vous voulez un meilleur retour sur les investissements que celui que vous procure votre système actuel.
- **Extension** : en commençant avec une infrastructure informatique existante, vous déployez le logiciel Java Enterprise System qui ne fait pas partie de votre système actuellement. En règle générale, vous procédez à l'extension de votre système logiciel de cette manière car vous devez satisfaire de nouveaux besoins de l'entreprise. Vous aurez peut-être besoin de nouvelles fonctionnalités, telles que l'agrégation personnalisée de services existants par le biais d'un portail Java Enterprise System ou l'autorisation et l'authentification Java pour les services existants.
- **Mise à niveau** : en commençant avec une infrastructure informatique composée d'une ancienne version de Java Enterprise System ou de produits Sun antérieurs à Java Enterprise System, vous effectuez la mise à niveau vers la version la plus récente des composants Java Enterprise System.

Chaque scénario d'adoption contient ses propres problèmes et défis. Quel que soit le scénario d'adoption qui caractérise votre cas, le processus de cycle de vie de la solution représenté à la [Figure 1-3](#) s'applique. Toutefois, selon le scénario d'adoption utilisé, les problèmes à traiter et les ressources requises pour investir dans les phases du cycle de vie peuvent varier.

Les problèmes suivants concernent généralement les scénarios d'adoption à des niveaux variables :

- **Migration** : l'amélioration ou la mise à niveau de l'infrastructure existante avec un nouveau logiciel requiert fréquemment la migration des données depuis le système existant vers le nouveau système. Ces données peuvent correspondre à des informations de configuration, des informations utilisateur ou des informations d'applications. Vous devrez peut-être également migrer la logique d'entreprise ou de présentation en raison des nouvelles interfaces de programmation.
- **Intégration** : l'ajout d'un logiciel dans un système existant ou le remplacement de sous-systèmes logiciels requiert fréquemment l'intégration de nouveaux composants logiciels dans les sous-systèmes restants. L'intégration peut impliquer le développement de nouvelles couches d'interface à l'aide de connecteurs J2EE ou d'adaptateurs de ressources qui reconfigurent les composants existants et qui implémentent les plans de transformation de données.
- **Formation** : presque tous les changements apportés à l'infrastructure impliquent une modification des procédures et connaissances informatiques. Votre service informatique doit avoir suffisamment de temps pour acquérir de nouvelles connaissances ou pour transmettre d'anciennes connaissances afin de prendre en charge les technologies Java Enterprise System.
- **Matériel** : lorsque vous remplacez ou améliorez un système ou un sous-système existant, les contraintes d'entreprise peuvent exiger une réutilisation du matériel existant. Selon le scénario d'adoption choisi, les ressources matérielles peuvent devenir un facteur important.

Le tableau suivant résume la nature des problèmes qui concernent chacun des scénarios d'adoption de Java Enterprise System.

Tableau 1-6 Problèmes liés aux différents scénarios d'adoption Java ES

Scénario d'adoption	Migration	Intégration	Formation	Matériel
Nouveau système	N'est pas un problème	Intégration de nouveaux composants relativement facile	Problème important en règle générale	Compromis entre les coûts de l'équipement et de main-d'œuvre.
Amélioration	Peut être un problème sérieux	Nécessité d'intégrer de nouveaux composants dans le système existant	Peut être un gros problème	Peut impliquer des contraintes considérables en raison de l'équipement existant

Tableau 1-6 Problèmes liés aux différents scénarios d'adoption Java ES (*suite*)

Scénario d'adoption	Migration	Intégration	Formation	Matériel
Extension	N'est en principe pas un problème	Nécessité éventuelle d'intégrer de nouveaux composants dans le système existant	Peut être un gros problème	Requiert généralement du matériel neuf avec les mêmes compromis que sur un nouveau système
Mise à niveau	Peut être un gros problème	Intégration des composants mis à niveau relativement facile	Problème relativement mineur	Problème relativement mineur

* L'utilisation de plusieurs ordinateurs puissants fait généralement augmenter les coûts de l'équipement tout en exigeant moins de ressources informatiques. L'utilisation de nombreux petits ordinateurs fait généralement diminuer les coûts de l'équipement tout en exigeant davantage de ressources informatiques.

Termes clés de ce chapitre

Cette section explique les principaux termes clés utilisés dans ce chapitre, en insistant sur la clarification des relations entre ces termes et sur leur mode d'utilisation dans le contexte de Java Enterprise System.

scénario d'adoption raison générale du déploiement du logiciel Java Enterprise System, caractérisant le système logiciel avec lequel vous commencez et l'objectif que vous tentez d'atteindre. Il existe quatre principaux scénarios d'adoption Java Enterprise System : nouveau système, remplacement, extension et mise à niveau.

composant unité de logique logicielle à partir de laquelle les applications distribuées sont créées. Un composant peut être un *composants système* inclus dans Java Enterprise System ou un *composant d'application* personnalisé. En règle générale, un composant d'application se conforme à un modèle de composant distribué (par exemple, CORBA et la plate-forme J2EE™) et il exécute une fonction informatique spécifique. Ces composants, qu'ils soient seuls ou associés, fournissent des *services d'entreprise* et ils peuvent être englobés sous la forme de *services Web*.

application d'entreprise distribuée application dont la logique se développe sur un environnement réseau ou Internet (l'aspect distribué) et dont l'étendue et l'échelle satisfont les besoins d'un environnement de production ou d'un fournisseur de services (l'aspect entreprise).

utilisateur final personne qui utilise une application distribuée, la plupart du temps par le biais d'une interface graphique utilisateur, telle qu'un navigateur Internet ou l'interface graphique d'un périphérique mobile.

Le nombre d'utilisateurs finals simultanés pris en charge par une application est un facteur déterminant de l'*architecture de déploiement* de l'application.

service fonction logicielle exécutée pour un ou plusieurs *clients*. Cette fonction peut se situer à un niveau très bas, tel que la gestion de la mémoire, ou à un niveau élevé, tel qu'un *services d'entreprise* de vérification de crédit. Un service de haut niveau peut se composer d'une famille de services. Les services peuvent être locaux (disponibles pour les clients locaux) ou distribués (disponibles pour les clients distants).

composant de qualité de service un des types de *composants système* inclus dans Java Enterprise System. Les composants de prise en charge, qui incluent les composants d'accès et les composants administratifs, assurent la prise en charge des *composants de service du système*.

composant partagé un des types de *composants système* inclus dans Java Enterprise System. Les composants partagés, en règle générale des bibliothèques, fournissent des services locaux aux autres composants du système. En revanche, un *composants de service du système* fournit des services d'infrastructure distribués aux autres composants du système (ou aux *composants d'application*).

composant système tout package ou ensemble de packages inclus dans Java Enterprise System et installé par le programme d'installation de Java Enterprise System. Il existe plusieurs types de composants système : les *composants de service du système* qui fournissent des *services* d'infrastructure distribués, les *composants de qualité de service* qui prennent en charge les composants de services du système en fournissant des services d'accès et administratifs et les *composants partagés* qui fournissent des services locaux aux autres composants système.

service système un ou plusieurs *services* distribués qui définissent la fonctionnalité unique fournie par Java Enterprise System. En principe, les services système requièrent la prise en charge de plusieurs *composants de qualité de service* ou de plusieurs *composants partagés*, voire de ces deux types de composants.

composant de service du système un des types de *composants système* inclus dans Java Enterprise System. Les composants des services système fournissent les principaux services d'infrastructure Java Enterprise System : services de portail, de communication et de collaboration, d'identité et de sécurité, Web et applicatifs, et de disponibilité.

Architectures de la solution Java Enterprise System

Ce chapitre présente les concepts architecturaux sur lesquels reposent les solutions Java Enterprise System (Java ES). Il explique comment les composants de Java ES (composants de service du système et composants de qualité de service) sont utilisés pour prendre en charge les solutions d'entreprise distribuées.

Les *architectures* de la solution Java ES possèdent deux aspects : une *architecture logique* et une *architecture de déploiement*. La première illustre les interactions entre les blocs fonctionnels logiques (composants logiciels) d'une solution, tandis que la seconde représente le mappage de l'architecture logique sur un environnement informatique physique. Les composants de Java ES jouent un rôle important au niveau des architectures logiques et de déploiement.

Ce chapitre décrit une structure architecturale destinée à la conception d'architectures de solution Java ES, suivie d'un exemple d'architecture s'appuyant sur cette structure.

Ce chapitre aborde les sujets suivants :

- « Structure architecturale de Java Enterprise System », page 34
- « Exemple d'architecture de solution Java Enterprise System », page 51
- « Termes clés de ce chapitre », page 54

Structure architecturale de Java Enterprise System

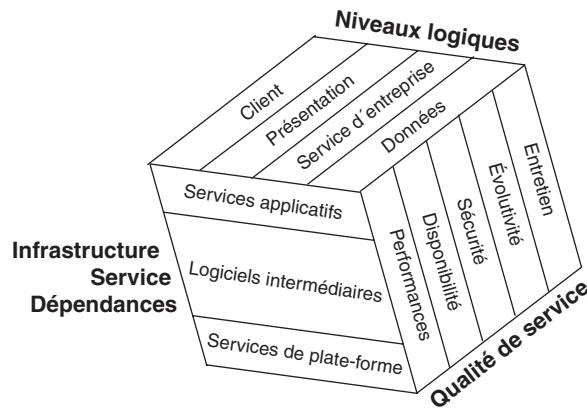
Les composants de Java Enterprise System prennent en charge le déploiement de solutions logicielles d'entreprise distribuées.

Pour satisfaire aux exigences de performances, de disponibilité, de sécurité, d'évolutivité et d'entretien requises par l'entreprise, les solutions logicielles doivent être correctement conçues.

Un certain nombre de paramètres entrent en jeu dans la conception de solutions logicielles d'entreprise distribuées. Ces paramètres représentent les différents aspects des interactions entre les composants logiciels utilisés pour la création de tels systèmes. La conception de systèmes distribués implique notamment les trois dimensions architecturales suivantes :

- **Dépendances des services d'infrastructure** : cette dimension souligne l'importance du rôle des composants de service du système (voir la section « [Composants de service du système](#) », page 21) dans la prise en charge des solutions distribuées.
- **Niveaux logiques** : cette dimension souligne l'importance de l'indépendance logique et physique des composants de solution dans le but de les déployer dans un environnement réseau ou Internet.
- **Qualité de service** : cette dimension insiste sur la manière dont les exigences de qualité de service, telles que la disponibilité, la sécurité, l'évolutivité et l'entretien, sont atteintes, y compris sur l'importance du rôle des composants de qualité de service (voir la section « [Composants de qualité de service](#) », page 23).

Ces trois dimensions sont illustrées à la figure suivante.

Figure 2-1 Dimensions d'une architecture de solution Java ES

La combinaison de ces trois dimensions représente une structure unique qui intègre les relations entre les composants logiciels (*composants d'application* et composants d'infrastructure) nécessaires à l'obtention des fonctions de service et de la qualité de service requises pour une solution logicielle.

Les sections suivantes décrivent ces trois dimensions individuellement, puis elles présentent une synthèse de ces trois dimensions sous la forme d'une structure unifiée.

Dimension 1 : Dépendances des services d'infrastructure

Les composants logiciels en interaction des applications d'entreprise distribuées requièrent un ensemble sous-jacent de services d'infrastructure afin que les composants distribués puissent communiquer entre eux, coordonner leur travail, implémenter un accès sécurisé, etc. Cette section explique le rôle essentiel joué par un certain nombre de composants de Java ES dans la prestation de ces services d'infrastructure.

Niveaux de services d'infrastructure

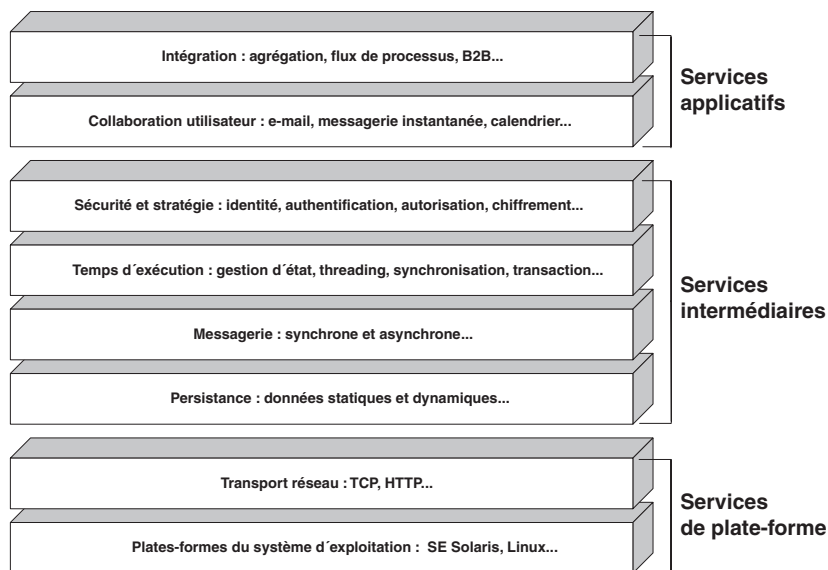
Lors de la conception d'un système logiciel distribué, qu'il s'agisse principalement de composants personnalisés ou de composants de Java ES prêts à l'emploi, vous devez intégrer un certain nombre de services d'infrastructure. Ces services fonctionnent sur plusieurs niveaux.

La dimension de dépendance des services d'infrastructure de l'architecture d'une solution est illustrée à la [Figure 2-2, page 36](#). Les niveaux représentés sur cette figure constituent une vue développée de la couche des services d'infrastructure décrits à la [Figure 1-1, page 19](#).

La hiérarchie des services décrits à la [Figure 2-2](#) et leurs dépendances constituent une dimension importante de l'architecture logique d'une solution. Ces services d'infrastructure fournissent la base conceptuelle pour comprendre le rôle des composants de service du système Java ES (voir la section « [Composants de service du système](#) », page 21).

En général, les services représentés à la [Figure 2-2](#) se divisent en trois grands groupes : des services de plate-forme de bas niveau, des services applicatifs de haut niveau et un groupe de logiciels intermédiaires, appelés ainsi du fait de leur emplacement entre les deux autres groupes.

Figure 2-2 Dimension 1 : Niveaux de services d'infrastructure



Les paragraphes suivants décrivent les différents niveaux de services d'infrastructure et se réfèrent aux artefacts du langage de programmation Java, le cas échéant. Les niveaux de services sont décrits en partant du niveau inférieur pour aller jusqu'au niveau supérieur, comme illustré à la [Figure 2-2](#) :

- **Plates-formes du système d'exploitation** : assure la prise en charge de base pour tout processus s'exécutant sur un ordinateur. Le système d'exploitation (tel que Solaris™, Linux ou Microsoft Windows) gère les périphériques physiques ainsi que la mémoire, les threads et les autres ressources requises pour la prise en charge de Java Virtual Machine (machine JVM™).
- **Transport réseau** : assure une prise en charge réseau de base pour la communication entre les composants d'applications distribuées s'exécutant sur différents ordinateurs. Ces services incluent la prise en charge des protocoles, tels que TCP et HTTP. Les autres protocoles de communication de niveau élevé (voir le niveau messagerie) dépendent de ces services de transport de base.
- **Persistance** : assure la prise en charge de l'accès et du stockage des données statiques (telles que les informations sur les utilisateurs, les répertoires ou la configuration) et les données d'application dynamiques (informations fréquemment mises à jour).
- **Messagerie** : assure la prise en charge de la communication synchrone et asynchrone entre les composants d'applications. La messagerie synchrone correspond à l'envoi et à la réception de messages en temps réel ; elle comporte également une fonction d'invocation de méthode distante (RMI) entre les composants J2EE et des interactions SOAP avec les services Web. La messagerie asynchrone, quant à elle, correspond à une communication pour laquelle l'envoi d'un message ne dépend pas de la capacité du destinataire à le recevoir immédiatement. Les spécifications de messagerie asynchrone, par exemple Java Message Service (JMS) et ebXML, prennent en charge une fiabilité garantie et d'autres sémantiques de messagerie.
- **Exécution** : assure la prise en charge requise par tout modèle de composant distribué, tel qu'un modèle J2EE ou CORBA. Outre l'invocation de méthode distante requise pour les composants distribués de couplage étroit, les services d'exécution incluent la gestion de l'état du composant (cycle de vie), la gestion de pools de threads, la synchronisation (verrouillage mutex), les services de persistance, le contrôle des transactions distribuées et le traitement des exceptions distribuées. Dans un environnement J2EE, ces services d'exécution sont fournis par des conteneurs EJB™, Web et des beans gérés par message dans un serveur d'applications ou un serveur Web.

- **Sécurité et stratégie** : assure la prise en charge de l'accès sécurisé aux ressources d'applications. Ces services incluent la prise en charge des stratégies qui régissent l'accès de groupe ou l'accès basé sur des rôles aux ressources distribuées ainsi que les fonctionnalités de *connexion unique*. La connexion unique permet que l'authentification d'un utilisateur sur un service d'un système distribué soit appliquée automatiquement aux autres services (composants J2EE, services métier et services Web) du système.
- **Collaboration utilisateur** : fournit des services qui jouent un rôle essentiel dans la prise en charge de la communication directe entre les utilisateurs et dans la prise en charge de la collaboration entre les utilisateurs d'environnements d'entreprise et Internet. Ces services sont des services d'entreprise de niveau application, fournis en principe par des serveurs autonomes (par exemple, un serveur de courrier ou un serveur de calendrier).
- **Intégration** : fournit des services qui regroupent les services d'entreprise existants. L'intégration fournit une interface commune qui permet d'accéder aux services comme dans un portail ou en intégrant ces services au moyen d'un moteur de processus qui les coordonne au sein d'un flux de production. L'intégration peut également avoir lieu sous forme d'interactions interentreprises entre différentes entreprises.

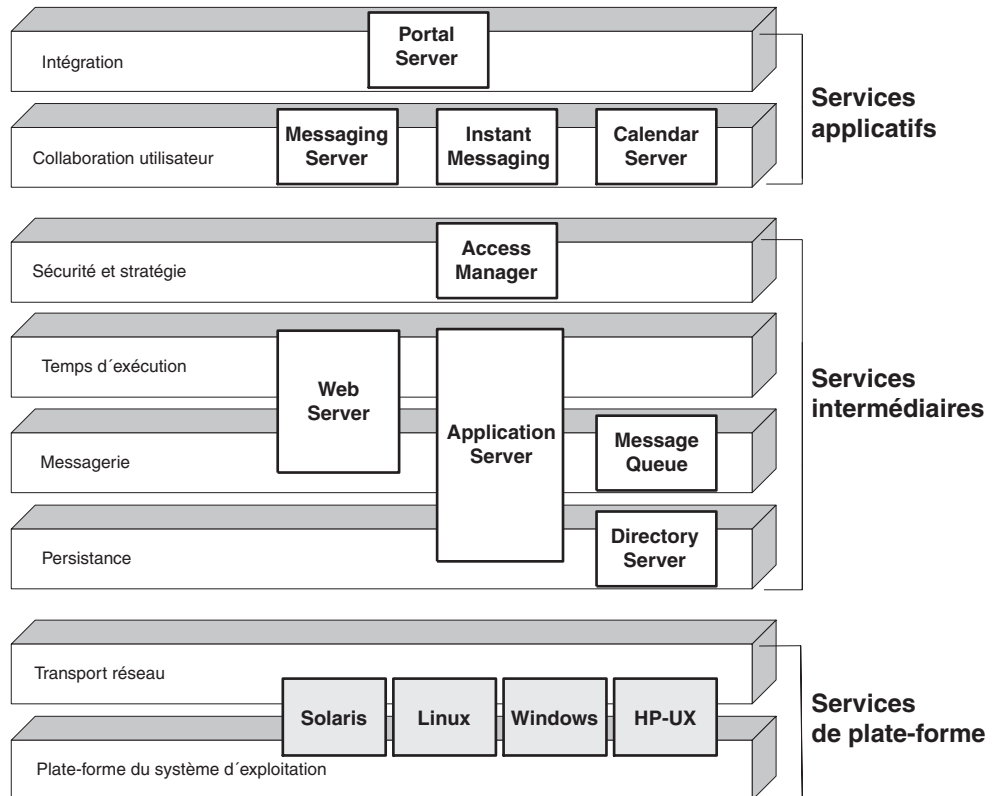
Les niveaux de services représentés à la [Figure 2-2](#) reflètent une interdépendance générale des divers services d'infrastructure, depuis les services de système d'exploitation de niveau inférieur jusqu'aux services applicatifs et services d'intégration de niveau supérieur. En règle générale, chaque service dépend des services situés en aval et prend en charge les services en amont.

Toutefois, la [Figure 2-2](#) ne représente pas une disposition en couches rigide des services d'infrastructure. Les services de niveau supérieur peuvent interagir directement avec les services de niveau inférieur sans dépendre des niveaux intermédiaires. Par exemple, certains services d'exécution peuvent dépendre directement des services de plate-forme sans avoir besoin des niveaux de service intermédiaires. De plus, d'autres niveaux de services, tels que le contrôle ou le service de gestion, peuvent être inclus dans cette illustration conceptuelle.

Composants des services d'infrastructure de Java Enterprise System

Les composants de Java ES implémentent les niveaux de services d'infrastructure distribués illustrés à la [Figure 2-2](#). Le positionnement des composants de service du système Java ES situés dans les différents niveaux est illustré à la [Figure 2-3](#).

Figure 2-3 Composants de service du système Java ES



REMARQUE Les plates-formes de système d'exploitation illustrées à la [Figure 2-3](#) ne font pas partie formellement de Java Enterprise System ; cependant, elles sont incluses pour présenter les plates-formes de système d'exploitation sur lesquelles les composants de Java ES sont pris en charge.

Dépendances des services d'infrastructure de Java Enterprise System

En règle générale, chaque composant de service du système Java ES illustré à la [Figure 2-3](#) dépend des composants situés au-dessous dans l'infrastructure et il prend en charge les composants qui sont au-dessus. Ces relations de dépendance et de prise en charge sont un facteur clé dans la conception d'architectures logiques.

Le [Tableau 2-1](#) présente les relations spécifiques entre les composants de service du système Java ES, répertoriés de haut en bas, tels qu'ils sont illustrés à la [Figure 2-3](#).

Tableau 2-1 Relations entre les composants de service du système Java ES

Composant	Dépend de	Prend en charge
Portal Server	Application Server ou Web Server Access Manager Directory Server S'il est configuré pour utiliser les canaux correspondants : Calendar Server Messaging Server Instant Messaging	
Messaging Server	Directory Server Access Manager (pour la connexion unique)	Calendar Server (pour les notifications par e-mail) Portal Server (pour le canal de messagerie)
Instant Messaging	Directory Server Access Manager (pour la connexion unique)	Portal Server (pour le canal de messagerie instantanée)
Calendar Server	Directory Server Messaging Server (pour le service de notification par e-mail) Access Manager (pour la connexion unique)	Portal Server (pour le canal de calendrier)
Access Manager	Application Server ou Web Server Directory Server	Portal Server S'il est configuré pour la connexion unique : Calendar Server Messaging Server Instant Messaging

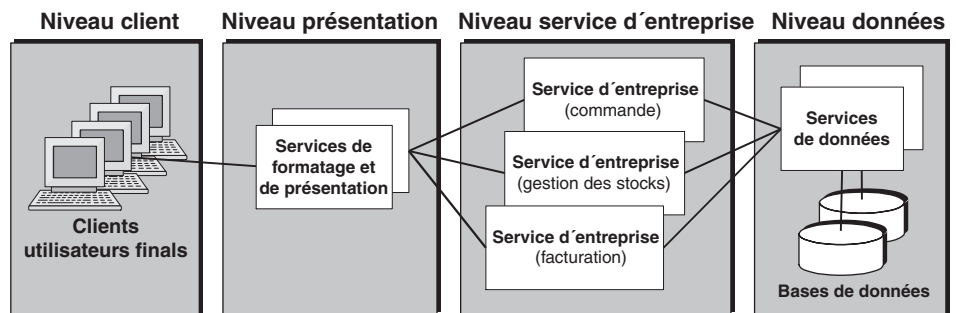
Tableau 2-1 Relations entre les composants de service du système Java ES (*suite*)

Composant	Dépend de	Prend en charge
Application Server	Message Queue Directory Server (pour les objets administrés)	Portal Server Access Manager
Message Queue	Directory Server (pour les objets administrés)	Application Server
Web Server	Access Manager (pour le contrôle d'accès)	Portal Server Access Manager
Directory Server	(aucune dépendance)	Portal Server Calendar Server Messaging Server Instant Messaging Access Manager

Dimension 2 : Niveaux logiques

Les composants logiciels en interaction des applications d'entreprise distribuées peuvent être considérés comme des éléments figurant dans plusieurs niveaux logiques. Ces niveaux représentent l'indépendance logique et physique des composants logiciels selon la nature des services qu'ils fournissent.

La dimension de niveau logique de l'architecture de la solution est illustrée à la figure suivante.

Figure 2-4 Dimension 2 : Niveaux logiques pour les applications d'entreprise distribuées

Pour la plus grande partie, les architectures des niveaux logiques représentent la couche des applications d'entreprise distribuées de la [Figure 1-1, page 19](#). Les composants de service du système Java ES traités dans la section « [Niveaux de services d'infrastructure](#) », [page 35](#) assurent la prise en charge des composants d'applications dans l'ensemble des niveaux logiques illustrés à la [Figure 2-4](#). Toutefois, les concepts de niveaux logiques s'appliquent également aux composants de service du système qui fournissent les services de niveau application, notamment Messaging Server et Calendar Server.

Description des niveaux logiques

Cette section présente une brève description des quatre niveaux logiques illustrés à la [Figure 2-4](#). Cette description concerne les composants d'applications implémentés à l'aide du modèle de composant de la plate-forme Java 2, Enterprise Edition (plate-forme J2EE™). Cependant, d'autres modèles de composants distribués, tels que CORBA, prennent également en charge cette architecture.

- **Niveau client** : le niveau client représente une logique d'application à laquelle un utilisateur final peut accéder directement par le biais d'une interface utilisateur. La logique du niveau client peut inclure les clients basés sur le navigateur, les composants Java s'exécutant sur un ordinateur de bureau ou les clients mobiles de la plate-forme Java 2, Micro Edition (plate-forme J2ME™) fonctionnant sur un périphérique de poche.
- **Niveau présentation** : le niveau présentation se compose de la logique d'application qui prépare les données pour les apporter au niveau client et traite les requêtes du niveau client pour les acheminer vers la logique d'entreprise d'arrière-plan. En principe, la logique du niveau présentation se compose des composants J2EE, tels que les composants Java Servlet ou JSP, qui préparent les données pour les acheminer au format HTML ou XML ou qui reçoivent les demandes de traitement. Ce niveau peut également comporter un service de portail capable d'assurer un accès personnalisé et sécurisé aux [services d'entreprise](#) du niveau service d'entreprise.

- **Niveau service d'entreprise** : le niveau service d'entreprise se compose d'une logique qui effectue les fonctions principales de l'application : traitement des données, implémentation des règles d'entreprise, coordination de plusieurs utilisateurs et gestion des ressources externes, telles que les bases de données et les systèmes existants. En principe, ce niveau comporte des composants de couplage étroit qui se conforment au modèle de composant distribué J2EE, tel que les objets Java, les composants EJB ou les beans gérés par message. Il est possible d'assembler les composants J2EE pour fournir des services d'entreprise complexes, tels qu'un service d'inventaire ou un service de calcul de taxe. Les composants individuels et les assemblages de services peuvent être regroupés sous forme de *services Web* de couplage flexible au sein d'un modèle d'architecture orienté service, qui se conforment aux normes d'interface SOAP (Simple Object Access Protocol, protocole SOAP). Il est également possible de créer des services d'entreprise sous forme de *serveurs* autonomes, tels qu'un serveur de messagerie ou un serveur de calendrier d'entreprise.
- **Niveau données** : le niveau données se compose de services qui fournissent des données persistantes pour la logique d'entreprise. Ces données peuvent correspondre à des données d'application stockées dans un système de gestion de base de données ou il peut s'agir d'informations de ressources et de répertoires stockées dans un magasin de données LDAP (Lightweight Directory Access Protocol, protocole LDAP). Les services de données peuvent également comporter des données provenant de sources externes ou des données accessibles à partir de systèmes informatiques existants.

Indépendance physique et logique

La dimension architecturale illustrée à la [Figure 2-4, page 41](#) met en évidence l'indépendance physique et logique des composants, représentée par quatre niveaux distincts. Ces niveaux expriment le partitionnement de la logique d'application sur les divers ordinateurs en réseau :

- **Indépendance logique** : les quatre niveaux du modèle architectural représentent l'indépendance logique. Vous pouvez modifier la logique d'application sur un niveau (par exemple, sur le niveau service d'entreprise) indépendamment de la logique sur les autres niveaux. Vous pouvez changer l'implémentation de la logique d'entreprise sans avoir à modifier ou à mettre à niveau la logique du niveau présentation ou du niveau client. Cette indépendance signifie, par exemple, que vous pouvez introduire de nouveaux types de composants clients sans avoir à modifier les composants de service d'entreprise.

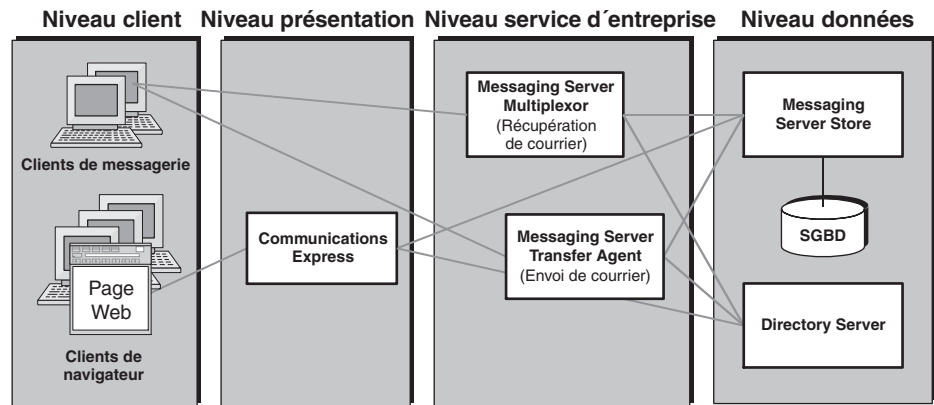
- **Indépendance physique** : les quatre niveaux représentent également l'indépendance physique. Vous pouvez déployer la logique dans différents niveaux sur diverses plates-formes matérielles (c'est-à-dire différentes configurations de processeur, divers jeux de puces et systèmes d'exploitation). Cette indépendance permet d'exécuter des composants d'applications distribuées sur les ordinateurs qui sont le mieux adaptés à leurs exigences individuelles et à l'optimisation de la bande passante réseau.

La manière dont vous mappez les composants de l'application ou de l'infrastructure sur un environnement matériel (c'est-à-dire votre architecture de déploiement) dépend de nombreux facteurs variant selon l'échelle et la complexité de votre solution logicielle. Pour de très petits déploiements, une architecture de déploiement peut comprendre un nombre réduit d'ordinateurs. Pour les déploiements à grande échelle, le mappage des composants sur un environnement matériel peut prendre en compte des facteurs, tels que la vitesse et la puissance des ordinateurs, la vitesse et la bande passante des liaisons réseau, les impératifs en termes de sécurité et de pare-feu, ainsi que les stratégies de réplication des composants pour une évolutivité et une disponibilité élevées.

Architecture à plusieurs niveaux appliquée aux composants système

Comme l'illustre la [Figure 2-3, page 39](#), les composants des services d'infrastructure Java ES assurent la prise en charge d'infrastructure sous-jacente pour les solutions logicielles distribuées. Toutefois, certaines de ces solutions englobent des services de niveau application fournis directement par les composants de Java ES. Ces solutions utilisent des approches de conception de niveau logique.

Par exemple, les services de communication par e-mail fournis par Messaging Server sont implémentés à l'aide de plusieurs configurations de Messaging Server logiquement distinctes. Chacune des configurations fournit un ensemble de services séparé. Lors de la conception de ces solutions de messagerie, ces configurations distinctes sont représentées sous forme de composants séparés situés sur différents niveaux logiques, comme illustré à la figure suivante.

Figure 2-5 Messaging Server : Exemple d'architecture à plusieurs niveaux

REMARQUE Le but de la [Figure 2-3](#) n'est pas de représenter une architecture logique complète. En effet, certains composants de Java ES ont été omis pour simplifier l'illustration. Les lignes reliant les composants représentent les interactions.

La séparation logique des fonctions de Messaging Server sur différents niveaux permet de déployer les configurations de Messaging Server logiquement distinctes sur différents ordinateurs d'un environnement physique. La séparation physique autorise une certaine flexibilité par rapport au respect des exigences de qualité de service (voir la section "[Dimension 3 : Qualité de service](#)"). Par exemple, elle fournit diverses solutions de disponibilité pour les différentes instances ainsi que diverses implémentations de sécurité pour les différentes fonctions de Messaging Server.

Dimension 3 : Qualité de service

Les deux dimensions architecturales précédentes (dépendances des services d'infrastructure et niveaux logiques) portent essentiellement sur les aspects logiques de l'architecture, c'est-à-dire les composants requis pour assurer l'interaction destinée à fournir les services appropriés aux utilisateurs finals. Cependant, il existe une dimension tout aussi importante pour une solution déployée : la capacité de la solution à satisfaire les exigences de qualité de service.

La dimension de la qualité de service dans l'architecture de la solution met en évidence le rôle joué par les composants de qualité de service de Java ES.

Qualités de service

Compte tenu que les services Internet et d'e-commerce ont pris davantage d'importance dans les opérations d'entreprise, les performances, la sécurité, l'évolutivité et l'entretien de ces services sont devenus des exigences clés en termes de qualité de service pour les architectures de déploiement à grande échelle et de haute performance.

Pour concevoir une solution logicielle réussie, vous devez établir des exigences de qualité de service pertinentes et concevoir une architecture qui satisfait ces exigences. Certaines qualités de service importantes sont utilisées pour spécifier les exigences de qualité de service. Ces qualités de service sont répertoriées dans le tableau suivant.

Tableau 2-2 Qualités de service affectant l'architecture de la solution

Qualités de service du système	Description
Performances	Mesure du temps de réponse et de la latence par rapport aux conditions de chargement de l'utilisateur.
Disponibilité	Mesure de la fréquence à laquelle les ressources et services d'un système sont accessibles aux utilisateurs finals (<i>temps d'activité</i> d'un système).
Sécurité	Combinaison complexe de facteurs décrivant l'intégrité d'un système et de ses utilisateurs. La sécurité implique la sécurité physique des systèmes, la sécurité du réseau, la sécurité des applications et des données (authentification et autorisation des utilisateurs) ainsi que le transport sécurisé des informations.
Évolutivité	Possibilité d'ajouter de la capacité à un système déployé dans le temps. En principe, l'évolutivité implique l'ajout de ressources au système, mais elle ne doit pas requérir de changements au niveau de l'architecture du déploiement.
Capacité latente	Aptitude d'un système à traiter une utilisation de charge de pointe inhabituelle sans ressources supplémentaires.
Entretien	Facilité avec laquelle un système déployé peut être entretenu. Cela comprend le contrôle du système, la réparation des problèmes se produisant et la mise à niveau des composants matériels et logiciels.

La dimension de la qualité de service affecte considérablement l'architecture de déploiement d'une solution, à savoir, la manière dont les composants d'applications et les composants d'infrastructure sont déployés dans un environnement physique.

Les qualités de service qui affectent l'architecture de déploiement sont intimement liées : les exigences inhérentes à une qualité du système ont souvent une influence sur la conception des autres qualités de service. Par exemple, des niveaux de sécurité relativement élevés peuvent affecter la performance qui, à son tour, est susceptible d'avoir une influence sur la disponibilité. L'ajout d'ordinateurs supplémentaires pour traiter les problèmes de disponibilité par la redondance affecte souvent les frais de maintenance (entretien).

Il est capital de comprendre la manière dont les qualités de service sont liées et de savoir quels compromis sont à faire pour concevoir des architectures de déploiement qui satisfont les exigences et les contraintes d'entreprise.

Composants de qualité de service de Java Enterprise System

Plusieurs composants de Java ES sont utilisés principalement pour améliorer la qualité des services fournis par les composants de service du système ou les composants d'applications distribuées. Ces composants logiciels sont fréquemment utilisés en association avec des composants matériels, tels que les équilibrateurs de charge et les pare-feu.

Les composants de qualité de service de Java ES, présentés à la section « [Composants de qualité de service](#) », page 23, sont résumés de la manière suivante :

- **Composants de disponibilité** : ces composants assurent un temps d'activité quasi continu d'une solution déployée.
- **Composants d'accès** : ces composants fournissent un accès Internet sécurisé aux services système ainsi qu'une fonction de routage.
- **Composants administratifs** : ces composants assurent un entretien amélioré des composants système.

Le tableau ci-dessous répertorie les composants de qualité de service de Java ES les plus importants d'un point de vue architectural avec les qualités système sur lesquelles ils ont la plus grande influence.

Tableau 2-3 Composants de qualité de service et qualités système influencées

Composant	Qualités système influencées
Communications Express	Sécurité Évolutivité
Directory Proxy Server	Sécurité Évolutivité
High Availability Session Store	Disponibilité
Portal Server Secure Remote Access	Sécurité Évolutivité
Sun Cluster	Disponibilité Évolutivité
Sun Remote Services Net Connect	Entretien

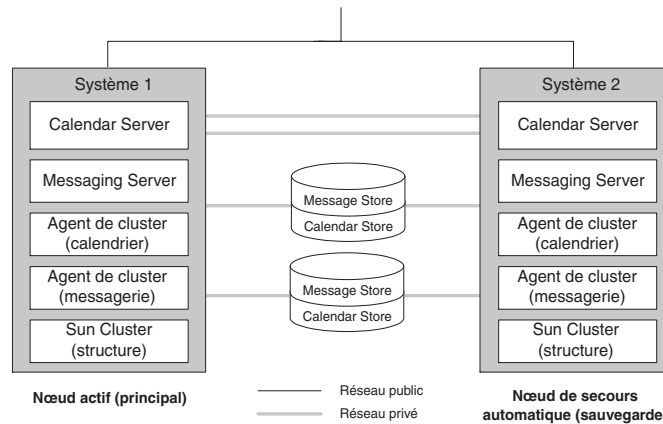
Logiciel Sun Cluster

Le logiciel Sun Cluster assure des services de disponibilité et d'évolutivité élevées pour les composants de Java ES et pour les applications prises en charge par l'infrastructure Java ES.

Un cluster est un ensemble d'ordinateurs de couplage flexible qui fournissent une vue client unique des services, des ressources système et des données. Au niveau interne, le cluster utilise les ordinateurs redondants, les interconnexions, le stockage de données et les interfaces réseau pour assurer une haute disponibilité aux données et aux services basés sur le cluster.

Le logiciel Sun Cluster contrôle en permanence l'état des nœuds membres et des autres ressources du cluster. En cas d'échec, Sun Cluster intervient pour initier le basculement des ressources qu'il contrôle en utilisant la redondance interne pour assurer un accès quasi continu à ces ressources.

La figure suivante présente un cluster à deux nœuds qui prend en charge les services du magasin de données pour Messaging Server et Calendar Server.

Figure 2-6 Conception de disponibilité à l'aide de nœuds Sun Cluster

Les packages de services de données Sun Cluster (parfois appelés agents Sun Cluster) sont disponibles pour l'ensemble des composants de service du système Java ES. Vous pouvez également écrire des agents pour les composants d'applications personnalisées.

En raison du contrôle effectué par le logiciel Sun Cluster, celui-ci peut également fournir des services évolutifs. Le renforcement du système de fichiers globaux d'un cluster et la capacité de plusieurs nœuds d'un cluster à exécuter des services d'infrastructure ou des services applicatifs permettent d'équilibrer la demande accrue de ces services sur de multiples instances simultanées. Si le logiciel Sun Cluster est configuré correctement, il peut assurer une disponibilité et une évolutivité élevées dans une application d'entreprise distribuée.

Du fait de la redondance nécessaire à la prise en charge des environnements Sun Cluster, l'intégration de Sun Cluster dans une solution fait considérablement augmenter le nombre d'ordinateurs et de liaisons réseau requis dans votre environnement physique.

Contrairement aux services fournis par les autres composants de Java ES, les services de disponibilité Sun Cluster sont des services distribués d'égal à égal. Par conséquent, le logiciel Sun Cluster doit être installé sur tous les ordinateurs d'un cluster.

Synthèse des trois dimensions architecturales

Lorsqu'elles sont considérées dans leur ensemble, les trois dimensions architecturales présentées à la [Figure 2-1](#) et abordées dans les sections précédentes fournissent une structure permettant de concevoir des solutions logicielles distribuées. Ces trois dimensions (dépendances des services d'infrastructure, niveaux logiques et qualité de service) soulignent le rôle joué par les composants de Java ES dans les architectures de solution.

Chaque dimension représente un aspect d'architecture spécifique. Toute architecture de solution doit prendre en compte la totalité de ces dimensions. Par exemple, les composants distribués de chaque niveau logique d'une architecture de solution (dimension 2) doivent être pris en charge par les composants d'infrastructure appropriés (dimension 1) et les composants de qualité de service adéquats (dimension 3).

De même, tout composant d'une architecture de solution joue des rôles différents par rapport aux diverses dimensions architecturales. Par exemple, Directory Server peut être considéré comme un composant d'arrière-plan du niveau données (dimension 2) et comme prestataire de services de persistance (dimension 1).

Du fait de la position centrale de Directory Server par rapport à ces deux dimensions, les problèmes de qualité de service (dimension 3) sont d'une importance capitale pour ce composant de Java ES. Une panne de Directory Server pouvant avoir un impact énorme sur le système d'entreprise, la conception de haute disponibilité est très importante pour ce composant. De plus, sachant que Directory Server est utilisé pour le stockage d'informations sensibles sur les utilisateurs ou la configuration, la conception de sécurité pour ce composant revêt également une importance capitale.

L'interaction de ces trois dimensions par rapport aux composants de Java ES influence la conception des architectures logiques et de déploiement de la solution.

Le but de ce manuel n'est pas de fournir une présentation détaillée des méthodes de conception selon la structure architecturale illustrée à la [Figure 2-1](#). Toutefois, la structure d'architecture tridimensionnelle souligne les aspects de conception importants pour comprendre le déploiement des solutions logicielles basées sur Java Enterprise System.

Exemple d'architecture de solution Java Enterprise System

Java Enterprise System prend en charge une large gamme de solutions logicielles.

La plupart des solutions peuvent être conçues et déployées immédiatement sans effort de développement à l'aide des composants inclus dans Java Enterprise System. D'autres solutions peuvent exiger des efforts de développement considérables, nécessitant le développement de composants J2EE personnalisés afin de fournir de nouveaux services d'entreprise ou de présentation. Vous pouvez englober ces composants personnalisés sous la forme de services Web conformes aux normes d'interface SOAP. La plupart de ces solutions impliquent une combinaison de ces deux approches.

Cette section comporte un exemple qui illustre comment Java Enterprise System prend en charge une solution prête à l'emploi, en s'inspirant des concepts architecturaux de la section précédente.

Scénario de communication d'entreprise

Généralement, les entreprises ont besoin de prendre en charge la communication entre leurs employés, en particulier les services de courrier et de calendrier. Ces entreprises pensent qu'il est avantageux pour leurs employés d'avoir un accès personnalisé aux sites Web internes et aux autres ressources basées sur les services d'autorisation et d'authentification de l'entreprise. De plus, ces entreprises souhaitent que l'identité des employés soit suivie dans tous les services de l'entreprise afin qu'une connexion Web unique permette d'accéder à ces services.

Ces exigences d'entreprise spécifiques, qui ne constituent qu'un exemple parmi d'autres, sont résumées dans le tableau ci-dessous.

Tableau 2-4 Résumé des exigences d'entreprise : scénario de communication

Exigence de l'entreprise	Description	Services Java ES requis
Connexion unique	Accès aux ressources d'entreprise sécurisées et aux services basés sur une identité unique avec une connexion unique pour l'accès Web.	Services d'identité

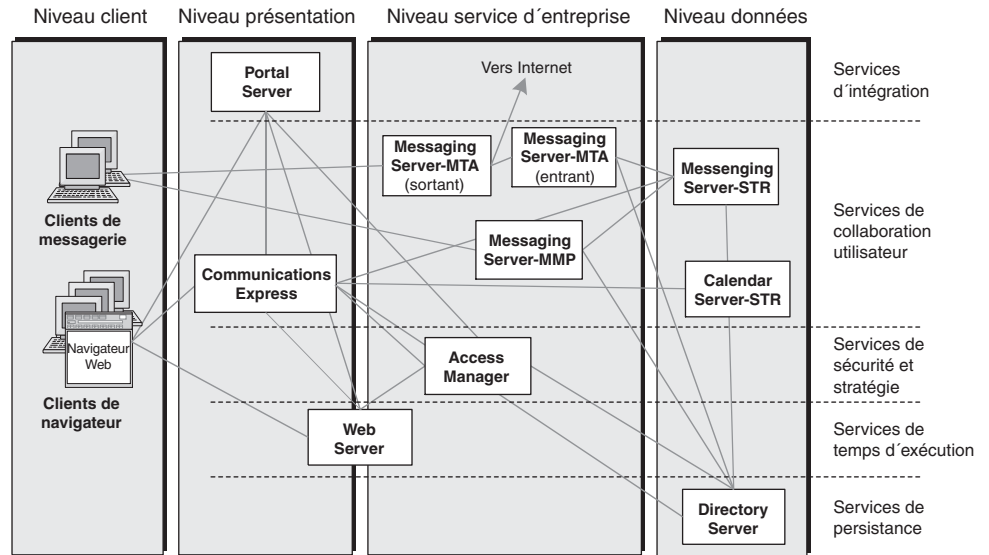
Tableau 2-4 Résumé des exigences d'entreprise : scénario de communication (*suite*)

Exigence de l'entreprise	Description	Services Java ES requis
Messagerie	Messagerie électronique assurant la communication entre les employés et le monde extérieur.	Services de communication et de collaboration
Calendrier	Dispositions électroniques des employés concernant le calendrier et les réunions.	
Accès au portail	Point d'accès Web unique et personnalisé aux services de communication, tels que l'e-mail, le calendrier et les pages Web internes.	Services de portail

En outre, toute entreprise doit faire face à des exigences de performances, de disponibilité, de sécurité réseau et d'évolutivité du système logiciel qui fournit ces services.

Architecture logique de l'exemple de scénario

La figure ci-après présente une architecture logique permettant de fournir les services de portail, de communication et d'identité répertoriés dans le [Tableau 2-4](#) à l'aide de composants de Java ES. L'architecture traite les configurations logiques spécifiques de Messaging Server comme composants séparés du fait des services distincts fournis par chacun d'entre eux.

Figure 2-7 Architecture logique d'un scénario de communication d'entreprise

Les composants placés dans une dimension horizontale représentent les niveaux logiques standard et ceux placés dans une dimension verticale représentent les niveaux de services d'infrastructure. Les interactions entre les composants dépendent de leurs fonctions comme services d'infrastructure distribués (interactions entre les niveaux de services d'infrastructure) ou de leurs rôles au sein d'une architecture d'application à plusieurs niveaux (interactions au sein des niveaux logiques et entre ceux-ci).

Dans cette architecture, Access Manager, qui accède aux informations utilisateur stockées dans Directory Server, est l'arbitre de l'autorisation et de l'authentification de connexion unique pour Portal Server et les autres composants Web du niveau de présentation. Les composants Messaging Server permettent le stockage des messages (Messaging Server-STR) au niveau données, l'envoi et la récupération des composants au niveau services d'entreprise, et fournissent un composant d'accès HTTP et Communications Express au niveau présentation.

L'architecture logique présente également les dépendances de services d'infrastructure entre les divers composants de Java ES. Par exemple, Portal Server dépend de Communications Express pour ses canaux de messagerie et de calendrier, alors qu'il dépend d'Access Manager pour les services d'authentification et d'autorisation. Ces composants dépendent, à leur tour, de Directory Server en ce qui concerne les informations utilisateur et les données de configuration. Certains composants nécessitent des services de conteneur Web fournis par Web Server.

Pour en savoir plus sur la conception logique de la solution Java ES, reportez-vous au *Guide de planification du déploiement de Sun Java Enterprise System 2005T1*.

Architecture de déploiement de l'exemple de scénario

Du fait du passage de l'architecture logique à une architecture de déploiement, les exigences de qualité de service deviennent essentielles. Par exemple, les pare-feu et les sous-réseaux protégés peuvent être utilisés pour créer une barrière de sécurité pour les données d'arrière-plan. Les exigences de disponibilité et d'évolutivité peuvent être satisfaites pour la plupart des composants en les déployant sur plusieurs ordinateurs et en utilisant des équilibrateurs de charge pour distribuer les requêtes parmi les composants répliqués.

Toutefois, lorsque des exigences de disponibilité plus contraignantes s'appliquent et lorsqu'une grande quantité de stockage sur disque est impliquée, d'autres solutions de disponibilité sont plus appropriées. Par exemple, Sun Cluster peut être utilisé pour le stockage de Messaging Server et la réplication multimaître peut être employée pour Directory Server.

Pour en savoir plus sur la conception du déploiement de la solution Java ES, reportez-vous au *Guide de planification du déploiement de Sun Java Enterprise System 2005T1*.

Termes clés de ce chapitre

Cette section explique les principaux termes clés utilisés dans ce chapitre, en insistant sur la clarification des relations entre ces termes et sur leur mode d'utilisation dans le contexte Java Enterprise System.

composant d'application *composant* logiciel personnalisé qui effectue une fonction informatique spécifique en fournissant des *services d'entreprise* aux *utilisateurs finals* ou aux autres composants d'applications. Un composant d'application se conforme en règle générale à un modèle de composant distribué (tel que CORBA et la plate-forme J2EE™). Ces composants, qu'ils soient seuls ou associés, peuvent être englobés sous la forme de *services Web*.

architecture conception présentant les blocs fonctionnels logiques et physiques d'une application distribuée (ou d'un autre système logiciel) et leurs relations. Dans le cas d'une *applications d'entreprise distribuées*, la conception architecturale inclut généralement l'*architecture logique* de l'application et l'*architecture de déploiement*.

service d'entreprise *composants d'application* ou assemblage de composants qui effectue la logique d'entreprise au nom de plusieurs clients (il s'agit donc d'un processus à unités d'exécution multiples). Un service d'entreprise peut également correspondre à un assemblage de composants distribués regroupés sous la forme d'un *services Web*, ou il peut s'agir d'un *serveur* autonome.

client logiciel demandant des *services* logiciels. (Remarque : il ne s'agit pas d'une personne, voir *utilisateurs finals*.) Un client peut correspondre à un service qui demande un autre service ou un composant d'interface graphique (IG) auquel un utilisateur final accède.

architecture de déploiement conception de haut niveau qui représente le mappage d'une *architecture logique* sur un environnement informatique physique. L'environnement physique inclut les ordinateurs d'un environnement intranet ou Internet, les liaisons réseau entre ceux-ci et tout autre périphérique physique requis pour prendre en charge le logiciel.

architecture logique conception qui représente les blocs fonctionnels logiques d'une application distribuée et les relations (ou interfaces) entre ces blocs fonctionnels. L'architecture logique inclut les *composants d'application* distribuées et les composants de services d'infrastructure requis pour les prendre en charge.

serveur processus logiciel à unités d'exécution multiples (différent d'un serveur matériel) qui fournit un *service* distribué ou un ensemble de services unis pour les *clients* qui accèdent au service au moyen d'une interface externe.

service Web service conforme aux protocoles Internet standardisés pour l'accessibilité, l'encapsulation de services et la découverte. Les normes incluent le protocole de messagerie SOAP (Simple Object Access Protocol), la définition d'interface WSDL (Web Service definition Language) et la norme de registre UDDI (Universal Discovery, Description, and Integration).

Termes clés de ce chapitre

Fonctions d'intégration de Java Enterprise System

Ce chapitre fournit des informations conceptuelles et techniques permettant de comprendre les fonctions qui jouent un rôle essentiel dans l'intégration de composants Java ES dans un système logiciel unique.

Ces fonctions permettent de mettre en évidence certains des avantages de l'utilisation de Java Enterprise System par rapport à l'intégration manuelle de différents produits d'infrastructure.

Ce chapitre traite des fonctions suivantes :

- « Le programme d'installation intégré de Java Enterprise System », page 58
- « Services intégrés d'identité et de sécurité », page 60
- « Termes clés de ce chapitre », page 65

Le programme d'installation intégré de Java Enterprise System

Tous les composants de Java ES sont installés au moyen d'un seul programme d'installation. Ce programme d'installation assure la cohérence des procédures et du comportement d'installation et de désinstallation pour tous les composants.

Le programme d'installation de Java ES est une structure intégrée qui transfère le logiciel Java ES sur un ordinateur hôte. Il permet de sélectionner et d'installer n'importe quel nombre de composants Java ES sur tout ordinateur de votre environnement informatique. Il assure également une configuration du temps d'installation selon les composants Java ES à installer.

Le programme d'installation de Java ES n'effectue pas lui-même d'installations distribuées. Pour déployer une solution logicielle Java ES distribuée, utilisez le programme d'installation de Java ES pour installer les composants appropriés sur chaque ordinateur de votre environnement, en procédant avec un ordinateur à la fois. Vous devez utiliser une séquence convenable de sessions d'installation et de procédures de configuration selon votre architecture de déploiement et les dépendances entre les composants.

Le programme d'installation est exécuté de manière interactive en mode graphique et en mode texte. Il comporte également un mode d'installation silencieux actionné par des paramètres. Outre l'anglais, le programme d'installation prend en charge sept langues : français, allemand, espagnol, coréen, chinois simplifié, chinois traditionnel et japonais.

Cette section aborde les aspects suivants du programme d'installation intégré de Java ES (pour de plus amples détails, reportez-vous au *Guide d'installation de Sun Java Enterprise System 2005T1*):

- [Contrôle de logiciels préexistants](#)
- [Contrôle de dépendance](#)
- [Configuration initiale](#)
- [Désinstallation](#)

Contrôle de logiciels préexistants

Le programme d'installation examine l'ordinateur sur lequel l'installation est effectuée, puis il identifie les composants Java ES déjà installés. Il procède ensuite à un contrôle sur plusieurs niveaux pour vérifier que tous les composants existants sont au niveau permettant une bonne interaction. Le programme d'installation signale les composants logiciels incompatibles qui doivent être mis à niveau ou supprimés.

De même, il recherche les composants partagés de Java ES (voir « [Composants partagés](#) », page 25) qui sont déjà installés, tels que J2SE ou NSS. Si le programme détecte des versions de composants partagés incompatibles, il les répertorie. Si vous effectuez l'installation, le programme met automatiquement à niveau les composants partagés.

Contrôle de dépendance

Le programme d'installation effectue un contrôle complet des composants afin de vérifier que les composants sélectionnés à installer fonctionnent correctement ensemble.

La plupart des composants ont des dépendances par rapport aux autres composants. Le programme d'installation comporte une logique qui garantit le respect de ces dépendances. De ce fait, lorsque vous sélectionnez un composant à installer, le programme d'installation inclut automatiquement les composants et les sous-composants par rapport auxquels le composant sélectionné a des dépendances.

Vous ne pouvez pas désélectionner un composant si un autre composant sélectionné en dépend au niveau local. Toutefois, si la dépendance n'est pas locale, vous recevez un avertissement, mais vous pouvez poursuivre en supposant que la dépendance est satisfaite par un composant figurant sur un autre ordinateur hôte.

Configuration initiale

La plupart des composants Java ES requièrent une configuration initiale avant de pouvoir être lancés. Pour certains composants, le programme d'installation de Java ES peut effectuer cette configuration initiale.

Vous pouvez choisir de laisser le programme d'installation effectuer cette configuration initiale (option Configurer maintenant) ou d'installer le logiciel sans effectuer la configuration initiale (option Configurer ultérieurement), auquel cas vous devrez configurer explicitement chaque composant installé une fois l'installation effectuée.

Si vous laissez au programme d'installation le soin d'effectuer la configuration initiale, indiquez les informations de configuration requises pendant l'installation. En particulier, vous pouvez indiquer un ensemble de paramètres communs à tous les composants, tels un ID d'administrateur et un mot de passe.

Désinstallation

Java Enterprise System comporte également un programme de désinstallation. Vous pouvez utiliser ce programme pour supprimer les composants qui ont été installés sur l'ordinateur local par le programme d'installation de Java ES. Le programme de désinstallation recherche les dépendances locales et émet des avertissements lorsqu'il détecte une telle dépendance. Le programme de désinstallation ne supprime pas les composants partagés de Java ES.

Tout comme le programme d'installation, le programme de désinstallation peut fonctionner en mode graphique, en mode texte ou en mode silencieux.

Services intégrés d'identité et de sécurité

La gestion intégrée des identités utilisateur et la structure intégrée d'authentification et d'autorisation sont deux des fonctions importantes de Java Enterprise System.

Les sections suivantes fournissent des informations techniques permettant de comprendre les services intégrés d'identité et de sécurité proposés par Java Enterprise System :

- [Identité unique](#)
- [Authentification et connexion unique](#)

Identité unique

Au sein d'un environnement Java ES, un utilisateur final possède une seule identité intégrée. Selon cette *identité unique*, un utilisateur peut être autorisé à accéder à diverses ressources, telles qu'un portail ou des pages Web, et à des services, tels que la messagerie, le calendrier et la messagerie instantanée.

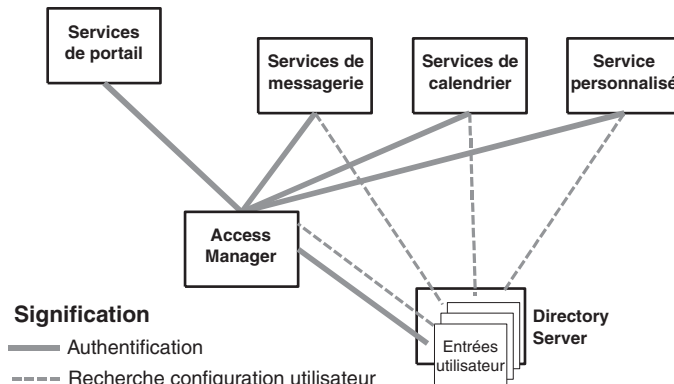
Cette fonction d'identité et de sécurité intégrées est basée sur une collaboration étroite entre Directory Server, Access Manager et les autres composants Java ES.

L'accès d'un utilisateur à un service ou à une ressource Java ES est obtenu en stockant des informations spécifiques à l'utilisateur dans une entrée utilisateur unique au sein d'un référentiel utilisateur ou d'un *répertoire*. En général, ces informations comportent les éléments suivants : nom et mot de passe uniques, adresse électronique, rôle au sein de l'organisation, préférences de pages Web, etc. Les informations de l'entrée utilisateur peuvent être utilisées pour authentifier l'utilisateur, autoriser son accès à des ressources spécifiques ou fournir divers services à cet utilisateur.

Dans le cas de Java Enterprise System, les entrées utilisateur sont stockées dans un répertoire fourni par Directory Server. Lorsqu'un utilisateur souhaite demander un service fourni par un composant Java ES, ce service utilise Access Manager pour authentifier l'utilisateur et autoriser l'accès aux ressources spécifiques. Le service demandé vérifie les informations de configuration propres à l'utilisateur qui se trouvent dans l'entrée du répertoire de l'utilisateur. Le service utilise ces informations pour effectuer le travail demandé par l'utilisateur.

La figure suivante illustre l'accès aux entrées utilisateur afin d'effectuer une authentification et une autorisation de l'utilisateur en vue de lui fournir des services.

Figure 3-1 Une entrée utilisateur unique prend en charge de nombreux services



L'une des fonctions dérivées de ce système est la capacité d'un utilisateur Web de se connecter à un service Java ESet, ce faisant, d'être authentifié automatiquement auprès des autres services du système. Cette capacité, appelée *connexion unique*, est une fonction puissante fournie par Java Enterprise System.

Authentification et connexion unique

Les services d'authentification et d'autorisation de Java ES sont fournis par Access Manager. Access Manager utilise les informations de Directory Server pour assurer l'interaction des utilisateurs avec les services Web de Java ES ou les autres services Web d'une entreprise.

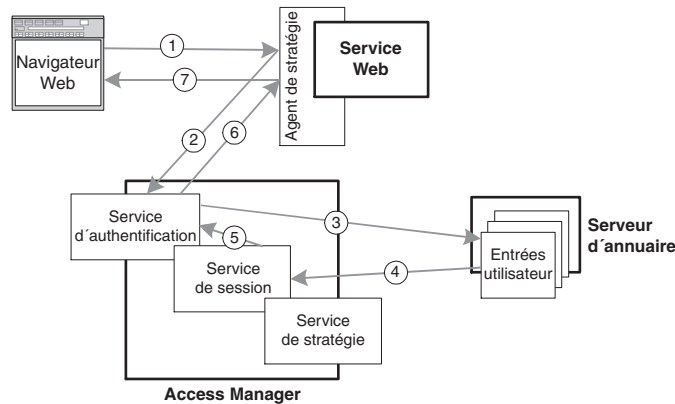
Access Manager utilise un composant externe appelé agent de stratégie. L'agent de stratégie se connecte au serveur Web qui héberge une ressource ou un service sécurisé par Access Manager. Il agit au nom d'Access Manager face aux demandes émises par les utilisateurs par rapport aux ressources sécurisées. Pour certains composants Java ES, tels que Portal Server et Communications Express, la fonctionnalité de l'agent de stratégie est assurée par un sous-composant d'Access Manager (voir « [Sun Java System Access Manager 6 2005T1](#) », page 80).

Authentification

Access Manager comporte un service d'authentification permettant de vérifier les identités des utilisateurs qui demandent l'accès (par HTTP ou HTTPS) aux services Web d'une entreprise. Par exemple, un employé qui recherche le numéro de téléphone d'un collègue utilise un navigateur pour aller sur l'annuaire en ligne de l'entreprise. Pour se connecter au service d'annuaire, l'utilisateur doit saisir un ID utilisateur et un mot de passe.

La séquence d'authentification est illustrée à la [Figure 3-2](#). Un agent de stratégie reçoit la demande de connexion à l'annuaire (1), puis il l'envoie au service d'authentification (2). Le service d'authentification vérifie l'ID utilisateur et le mot de passe en les comparant aux informations stockées dans Directory Server (3). Si la demande de connexion est valide, l'utilisateur est authentifié (4), (5) et (6), puis l'annuaire de l'entreprise s'affiche (7). Dans le cas contraire, une erreur est générée et l'authentification échoue.

Le service d'authentification prend également en charge l'authentification par certificats via HTTPS.

Figure 3-2 Séquence d'authentification

Connexion unique

Le scénario d'authentification traité dans les paragraphes précédents dissimule une étape importante. Lors de la vérification d'une demande d'authentification de l'utilisateur, le service de session d'Access Manager est activé (4), comme illustré à la [Figure 3-2](#). Le service de session génère un jeton de session qui détient les informations sur l'identité de l'utilisateur et un ID de jeton (5). Le jeton de session est renvoyé à l'agent de stratégie (6) qui transfère le jeton (sous la forme d'un cookie) vers le navigateur (7) depuis lequel la demande d'authentification a été effectuée.

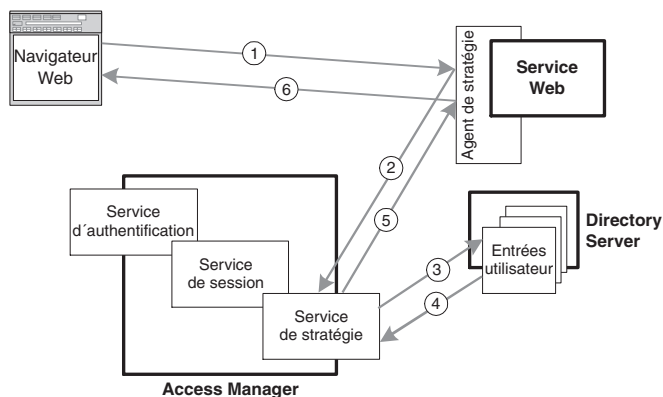
Si l'utilisateur authentifié essaie d'accéder à un autre service sécurisé, le navigateur transfère le jeton de session à l'agent de stratégie correspondant. Ce dernier vérifie avec le service de session que l'authentification précédente de l'utilisateur est toujours valide, puis l'utilisateur est autorisé à accéder au deuxième service sans avoir à entrer de nouveau son ID utilisateur et son mot de passe.

Par conséquent, il suffit à l'utilisateur de se connecter une seule fois pour être authentifié sur plusieurs services Web fournis par Java Enterprise System. L'authentification avec connexion unique reste valide jusqu'à la déconnexion de l'utilisateur ou jusqu'à l'expiration de la session.

Autorisation

Access Manager comporte également un service de stratégie qui assure un contrôle de l'accès aux ressources Web d'un environnement Java ES. Une *stratégie* est une règle décrivant les personnes autorisées à accéder à une ressource spécifique dans des conditions données. La séquence d'autorisation est illustrée à la figure suivante.

Figure 3-3 Séquence d'autorisation



Lorsqu'un utilisateur authentifié demande l'accès à une ressource sécurisée avec Access Manager (1), l'agent de stratégie avertit le service de stratégie (2) qui utilise les informations de Directory Server (3) pour évaluer la stratégie d'accès s'appliquant à la ressource et vérifier si l'utilisateur possède les droits d'accès à cette ressource (4). Si l'utilisateur possède des droits d'accès (5), la demande est satisfaite (6).

Access Manager permet de définir, de modifier, d'accorder, de révoquer et de supprimer les stratégies d'une entreprise. Les stratégies sont stockées dans Directory Server et configurées par des attributs liés aux stratégies dans les entrées d'organisation. Il est également possible de définir des rôles pour les utilisateurs et de les intégrer dans les définitions de stratégies.

Les agents de stratégie d'Access Manager sont chargés de la mise en œuvre des stratégies. Lorsque le service de stratégie rejette une demande d'accès, l'agent de stratégie empêche l'utilisateur d'accéder aux ressources sécurisées.

Termes clés de ce chapitre

Cette section explique les principaux termes clés utilisés dans ce chapitre, en insistant sur la clarification des relations entre ces termes et sur leur mode d'utilisation dans le contexte Java Enterprise System.

répertoire Type spécial de base de données optimisée pour la lecture des données et non pour leur écriture. La plupart des répertoires se basent sur le protocole standard LDAP (Lightweight Directory Access Protocol).

stratégie Règle décrivant les personnes autorisées à accéder à une ressource spécifique dans des conditions données. Cette règle peut se baser sur des groupes d'utilisateurs ou des rôles au sein d'une organisation.

identité unique Identité possédée par un utilisateur, car une seule entrée utilisateur se trouve dans un répertoire Java Enterprise System. Selon cette entrée utilisateur unique, un utilisateur peut être autorisé à accéder à diverses ressources Java Enterprise System, telles qu'un portail ou des pages Web, et à des services, tels que la messagerie, le calendrier et la messagerie instantanée.

connexion unique Fonction permettant l'authentification d'un utilisateur auprès d'un service d'un système distribué de manière que cette authentification soit automatiquement appliquée aux autres services du système.

Termes clés de ce chapitre

Tâches du cycle de vie de la solution Java Enterprise System

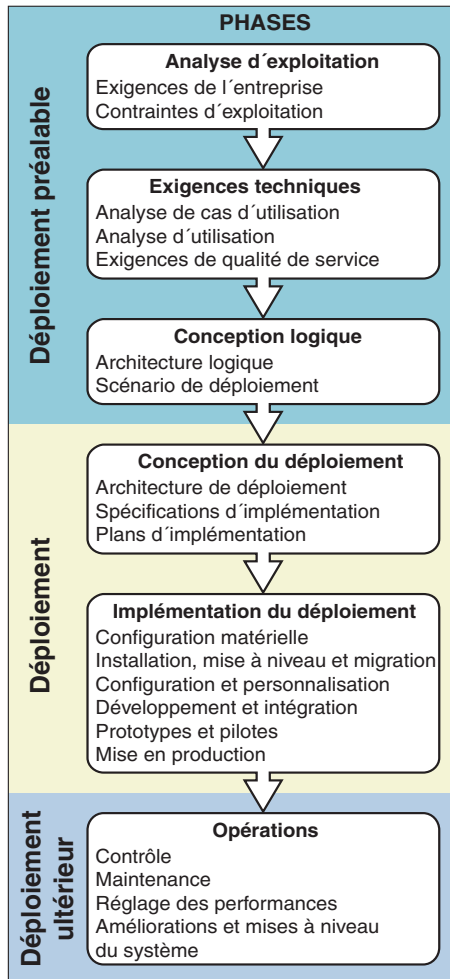
Le cycle de vie de la solution Java ES a été présenté au [chapitre 1, « Introduction à Java Enterprise System »](#), comme approche standard de mise en œuvre de solutions d'entreprise à l'aide du logiciel Java ES. Ce chapitre décrit les tâches impliquées dans chaque phase du cycle de vie. Le diagramme du cycle de vie est illustré à la [Figure 4-1, page 68](#) pour que vous puissiez vous y reporter plus facilement.

Ce chapitre aborde les concepts et la terminologie s'appliquant à chaque phase. Il traite essentiellement des tâches de déploiement, notamment des tâches de conception et de mise en œuvre du déploiement.

Ce chapitre s'articule autour des trois groupes de phases de cycle de vie ci-dessous :

- « [Déploiement préalable](#) », page 69
- « [Déploiement](#) », page 70
- « [Déploiement ultérieur](#) », page 76
- « [Termes clés de ce chapitre](#) », page 77

Figure 4-1 Tâches du cycle de vie d'une solution



Déploiement préalable

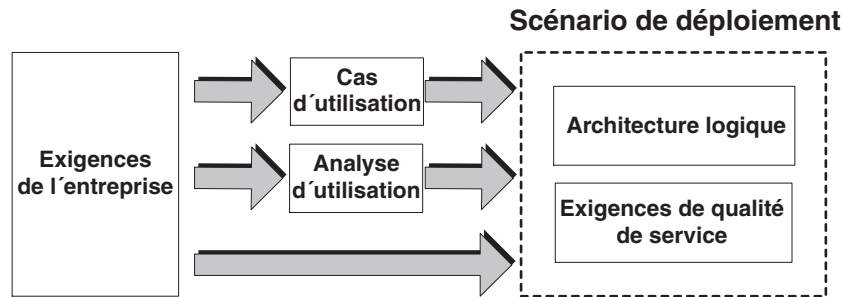
Au cours des phases de *déploiement préalable* du cycle de vie, vous convertissez une analyse des besoins de votre entreprise en un *scénario de déploiement*. Le scénario de déploiement fait office de spécification pour une conception de déploiement.

Les tâches de déploiement préalable sont regroupées en trois phases, comme illustré à la [Figure 4-1](#):

- **Analyse d'exploitation** : au cours de cette phase, vous définissez les objectifs d'exploitation d'un effort de déploiement proposé et stipulez les exigences et les contraintes de votre entreprise qui doivent être respectées pour atteindre ces objectifs.
- **Exigences techniques** : au cours de cette phase, vous utilisez les résultats de l'analyse d'exploitation pour créer les *cas d'utilisation* qui conditionnent l'interaction utilisateur avec un système logiciel anticipé. Vous déterminez également les modèles d'utilisation attendus pour ces cas d'utilisation. À l'aide de l'analyse d'exploitation et de l'analyse d'utilisation, vous formulez des exigences en termes de qualité de service (voir le [Tableau 2-2, page 46](#)) que le déploiement proposé doit satisfaire.
- **Conception logique** : au cours de cette phase, vous analysez les cas d'utilisation développés durant la phase d'exigences techniques afin de déterminer les composants d'infrastructure Java ES et ceux des applications personnalisées qui sont requis pour fournir les services d'utilisateur final. À l'aide des concepts traités dans le [chapitre 2, « Architectures de la solution Java Enterprise System »](#), vous concevez une architecture logique. L'architecture logique présente tous les composants et toutes les interactions entre les composants qui sont nécessaires à la réalisation des cas d'utilisation d'une solution logicielle donnée.

L'architecture logique, associée aux performances, à la disponibilité, à la sécurité et aux autres exigences de qualité de service, est englobée dans un scénario de déploiement, comme illustré à la figure suivante. Pour en savoir plus sur les phases de déploiement préalable du cycle de vie, reportez-vous au *Guide de planification du déploiement de Sun Java Enterprise System 2005T1*.

Figure 4-2 Spécification d'un scénario de déploiement



Déploiement

Lors des phases de *déploiement* du cycle de vie, vous convertissez un scénario de déploiement en une conception de déploiement que vous implémentez, testez et mettez en production.

Le processus de déploiement englobe généralement les composants logiciels de tous les niveaux et de tous les niveaux de service d'infrastructure requis pour la prise en charge d'une solution logicielle. En règle générale, vous devrez déployer les composants d'applications personnalisées (composants J2EE, services Web ou autres serveurs) ainsi que les composants Java ES requis pour prendre en charge la solution.

Les tâches de déploiement sont regroupées en deux phases, comme illustré à la [Figure 4-1](#):

- **Conception du déploiement** : la conception du déploiement dépend de l'architecture logique d'une solution ainsi que des performances, de la disponibilité, de la sécurité, de l'évolutivité, de l'entretien et d'autres exigences de qualité de service qu'une solution doit satisfaire. La dimension de la qualité de service dans l'architecture de déploiement joue un rôle capital dans la phase de conception du déploiement.

- **Mise en œuvre du déploiement** : la mise en œuvre d'une conception de déploiement est un processus fréquemment itératif qui implique une configuration matérielle, une installation et une configuration logicielle, un développement et une intégration, des tests et d'autres aspects de la mise en production.

Les sections suivantes abordent de manière plus détaillée ces deux phases du processus de déploiement.

Conception du déploiement

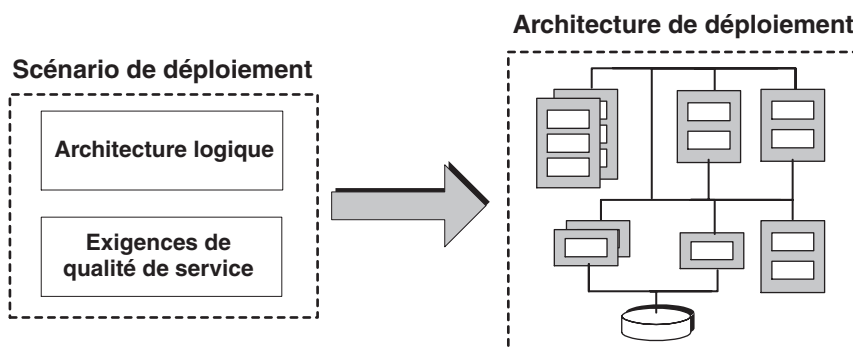
Lors de la phase de conception du déploiement, vous créez une architecture de déploiement de haut niveau, suivie de spécifications d'implémentation de bas niveau.

Architecture de déploiement

Une architecture de déploiement est créée en mappant les blocs fonctionnels logiques d'une application (l'architecture logique) sur un environnement informatique physique de sorte que les exigences de qualité de service spécifiées dans le scénario de déploiement soient respectées.

Le scénario de déploiement est converti en une architecture de déploiement, comme illustré à la figure suivante.

Figure 4-3 Conversion d'un scénario de déploiement en une architecture de déploiement



L'un des aspects de cette conception architecturale est le dimensionnement de l'environnement physique (en déterminant le nombre d'ordinateurs et en estimant la puissance de leurs processeurs et les exigences de RAM) pour satisfaire les performances, la disponibilité, la sécurité et les autres exigences de qualité de service. Une fois le dimensionnement effectué, mappez les composants Java ES et les composants d'applications sur les différents ordinateurs de l'environnement physique. L'architecture de déploiement produite doit prendre en compte les fonctionnalités des différents ordinateurs, les caractéristiques des services d'infrastructure système et les restrictions liées au coût total de possession ou de disponibilité.

Plus le nombre de composants Java ES du scénario de déploiement est élevé, plus vos exigences en termes de qualité de service seront contraignantes, et plus votre conception sera difficile sur des ordinateurs puissants et sur une bande passante réseau élevée. Si le matériel est limité ou si son prix est prohibitif, vous devrez peut-être envisager des compromis entre les frais fixes (matériel) et les frais variables (ressources humaines requises) ou entre les différentes exigences de qualité de service ; vous pourrez également intégrer des composants plus sophistiqués dans votre conception.

La conception d'une architecture de déploiement évolue fréquemment de manière itérative. Toutefois, comme point de départ de la conception de déploiement, Java Enterprise System développe un ensemble d'*architectures de déploiement de référence*.

Une architecture de référence est basée sur un scénario de déploiement spécifique : une architecture logique avec des exigences spécifiques en termes de qualité de service. Dans l'architecture de référence, une solution logicielle est déployée dans un environnement physique spécifique afin de répondre aux exigences de qualité de service spécifiées. Le test des performances sur des charges spécifiées repose sur le même ensemble de cas d'utilisation à partir desquels le scénario de déploiement a été développé. Les clients de Java ES peuvent accéder à la documentation relative à l'architecture de référence, à condition qu'ils ne la divulguent pas.

À partir d'une ou de plusieurs architectures de déploiement de référence, vous pouvez concevoir une première ébauche de l'architecture de déploiement qui satisfait vos propres exigences en matière de scénario de déploiement. Vous pouvez modifier les architectures de référence ou les utiliser comme points de référence en tenant compte de la différence entre votre propre scénario de déploiement et ceux sur lesquels reposent les architectures de référence. Ainsi, vous pouvez évaluer l'impact de vos propres exigences en termes de dimensionnement, de performances, de sécurité, de disponibilité, de capacité et d'entretien.

Spécifications d'implémentation

Les spécifications d'implémentation fournissent les informations nécessaires à la mise en œuvre d'une architecture de déploiement. Parmi ces spécifications, on compte généralement les suivantes :

- matériel utilisé, y compris les ordinateurs, les périphériques de stockage, les équilibreurs de charge et le câblage réseau ;
- systèmes d'exploitation ;
- conception réseau, notamment les sous-réseaux et les zones de sécurité ;
- détails de conception de disponibilité ;
- détails de conception de sécurité ;
- informations de conception de répertoire requises pour la configuration des utilisateurs finaux.

Plans d'implémentation

Les plans d'implémentation expliquent comment vous envisagez d'effectuer les diverses tâches lors de la phase de mise en œuvre du déploiement. Les plans abordent en règle générale les tâches suivantes :

- configuration matérielle ;
- installation logicielle, mise à niveau et migration ;
- configuration système et personnalisation ;
- développement et intégration ;
- tests ;
- mise en production.

Mise en œuvre du déploiement

La mise en œuvre d'une conception de déploiement regroupe les tâches énumérées dans la section précédente et présentées à la [Figure 4-1](#). Le processus de déploiement étant itératif par nature, l'ordre dans lequel les tâches sont effectuées n'a pas d'importance. Les sous-sections suivantes décrivent chacune des tâches principales de la mise en œuvre du déploiement dans l'ordre selon lequel elles sont généralement effectuées. Pour plus d'informations sur ces tâches, reportez-vous au document *Présentation de la documentation de Sun Java Enterprise System 2005T1*.

Configuration matérielle

La spécification d'implémentation inclut l'ensemble des détails de votre environnement physique : les ordinateurs, la conception du réseau, le matériel du réseau (y compris le câblage, les commutateurs, les routeurs et les équilibres de charge). L'ensemble de ce matériel doit être configuré comme plate-forme prenant en charge votre solution Java ES.

Installation logicielle, mise à niveau et migration

L'architecture de déploiement ainsi que les détails supplémentaires fournis dans les spécifications d'implémentation indiquent les composants d'applications et les composants Java ES qui doivent être installés sur chaque ordinateur de votre environnement physique. Vous pouvez utiliser le programme d'installation intégré de Java ES afin d'installer les composants appropriés sur chaque ordinateur de votre architecture de déploiement (voir la section « [Le programme d'installation intégré de Java Enterprise System](#) », page 58).

Le plan de votre installation décrit la séquence et l'étendue des sessions du programme d'installation. Toutefois, l'approche adoptée pour effectuer l'installation varie selon qu'il s'agit d'une nouvelle installation de Java Enterprise System, d'une mise à niveau de composants Java ES déjà installés ou d'un remplacement de composants tiers par Java Enterprise System. Les deux derniers scénarios d'installation de Java ES requièrent souvent une migration des données ou du code d'application pour garantir la compatibilité.

Configuration système et personnalisation

Vous devez effectuer un certain nombre de tâches de configuration système pour que les divers composants système fonctionnent en parfaite harmonie. Parmi ces tâches, la première consiste à effectuer la configuration initiale requise pour que chaque composant système puisse être lancé. Ensuite, chaque composant Java ES doit être configuré pour pouvoir communiquer avec les composants avec lesquels il interagit.

La haute disponibilité doit également être configurée selon la solution de disponibilité de chaque composant. Les utilisateurs doivent être configurés pour pouvoir accéder aux divers services ; les stratégies et contrôles d'authentification et d'autorisation doivent l'être également (voir la section « [Services intégrés d'identité et de sécurité](#) », page 60).

Dans la plupart des cas, les tâches de configuration impliquent un certain degré de personnalisation des composants de Java ES afin d'obtenir toutes les fonctions dont vous avez besoin. Par exemple, vous personnalisez Portal Server pour fournir des canaux de portail, Access Manager pour effectuer des tâches d'autorisation et Messaging Server pour utiliser le contrôle des virus et le filtrage du courrier indésirable.

Développement et intégration

L'architecture logique spécifiée dans le scénario de déploiement détermine généralement l'étendue du travail de *développement* personnalisé nécessaire à la mise en œuvre d'une solution.

Pour certains déploiements, le développement peut être relativement approfondi et nécessiter le développement de nouveaux services d'entreprise et de présentation à l'aide de composants J2EE s'exécutant sous un environnement Application Server ou Web Server. Dans ce cas, vous devez élaborer une première solution et démontrer le bien-fondé de sa conception avant d'entreprendre son développement complet.

Pour les solutions nécessitant un développement approfondi, Sun Java Studio procure des outils permettant de programmer les services d'entreprise ou les composants distribués. Il simplifie la programmation et les tests des applications prises en charge par l'infrastructure Java ES.

Dans certains cas, les composants de Java ES peuvent être intégrés à d'anciennes applications ou à des services tiers. Si tel est le cas, des répertoires ou des services de données doivent exister au niveau des données, ou des composants au niveau des services d'entreprise. L'intégration de composants Java ES à ces systèmes peut nécessiter la migration des données ou du code d'application.

La plate-forme J2EE fournit une structure de connecteurs qui permet de relier les applications existantes sous Application Server en développant les adaptateurs de ressources J2EE, tandis que Message Queue procure une fonctionnalité performante de messagerie asynchrone pour l'intégration d'applications diverses.

Test des prototypes et des pilotes

Selon le niveau de personnalisation ou de développement requis, vous devrez à un certain moment vérifier votre architecture de déploiement ; vous devrez tester la solution par rapport aux cas d'utilisation afin de vérifier que les exigences de qualité de service peuvent être satisfaites.

Si vous possédez peu de services personnalisés (déploiement relativement simple), la personnalisation des composants de Java ES et un test pilote du système peuvent suffire.

Toutefois, si vous avez développé une logique d'application relativement nouvelle et si vous avez créé des services personnalisés, la phase de tests peut s'avérer plus importante (tests de prototype, tests d'intégration, etc.).

Si ces tests mettent en évidence des points faibles dans votre architecture de déploiement, modifiez l'architecture et effectuez de nouveaux tests. Au bout du compte, ce processus itératif permet de procéder à l'implémentation et au déploiement d'une architecture dans un environnement de production.

Mise en production

La mise en production implique l'intégration de votre déploiement dans un environnement de production. Cette phase implique l'installation, la configuration et le lancement de services d'infrastructure et d'applications distribuées dans un environnement de production, la création d'utilisateurs finaux du système de production, la configuration de stratégies d'accès et de connexion unique, etc. En règle générale, vous commencez par effectuer un déploiement restreint avant d'implémenter la solution à toute votre organisation. Au cours de ce processus, vous effectuez des essais au cours desquels vous appliquez des charges croissantes pour confirmer que les exigences de qualité de service sont satisfaites.

Déploiement ultérieur

Au cours de la phase de *déploiement ultérieur* du cycle de vie, vous exécutez une solution déployée dans un environnement de production. Les tâches suivantes jouent un rôle dans la phase d'opérations du cycle de vie :

- **Contrôle** : ces tâches incluent le contrôle régulier des performances et des fonctions du système.
- **Maintenance** : ces tâches incluent les fonctions administratives quotidiennes, telles que l'ajout de nouveaux utilisateurs finaux à un système, le changement de mot de passe, l'ajout de nouveaux utilisateurs administratifs, le changement des droits d'accès, la réalisation de sauvegardes régulières, etc.
- **Réglage des performances** : ces tâches incluent l'utilisation des informations de contrôle régulier pour déceler des goulots d'étranglement dans les opérations du système et tenter de les éliminer en modifiant les propriétés de configuration, en ajoutant des capacités, etc.

- Améliorations et mises à niveau du système : ces tâches incluent l'ajout de nouveaux composants Java ES sur un système afin d'intégrer une nouvelle fonctionnalité ou de remplacer les composants autres que les composants Java ES. Dans les deux cas, ces modifications peuvent entraîner une nouvelle conception du système, en commençant par les phases initiales du cycle de vie de la solution. Les tâches de mise à niveau sont plus limitées ; il s'agit généralement de mises à niveau de composants de Java ES.

Chaque composant de Java ES possède ses propres outils d'administration pour configurer, ajuster ou gérer ses opérations. L'objectif est de fournir une infrastructure de gestion et de contrôle commune et des outils d'administration permettant de gérer le système dans son ensemble.

Termes clés de ce chapitre

Cette section explique les principaux termes clés utilisés dans ce chapitre, en insistant sur la clarification des relations entre ces termes et sur leur mode d'utilisation dans le contexte Java Enterprise System.

déploiement phase du processus de cycle de vie de la solution Java Enterprise System au cours de laquelle un scénario de déploiement est converti en une conception de déploiement, mis en œuvre, prototypé et lancé dans un environnement de production. Le produit final de ce processus porte également le nom de déploiement (ou solution déployée).

scénario de déploiement *architecture logique* d'une solution Java Enterprise System et exigences de qualité de service que la solution doit respecter pour satisfaire les besoins de l'entreprise. Les exigences de qualité de service sont les suivantes : performances, disponibilité, sécurité, entretien et évolutivité/capacité latente. Un scénario de déploiement constitue le point de départ d'une conception de déploiement.

développement tâche du processus de déploiement de la solution Java Enterprise System au cours de laquelle les composants personnalisés d'une *architecture de déploiement* sont programmés et testés.

déploiement préalable phase du processus de cycle de vie de la solution Java Enterprise System au cours de laquelle les besoins de l'entreprise sont convertis en un *scénario de déploiement* (*architecture logique* et ensemble d'exigences de qualité de service qu'une solution doit satisfaire).

déploiement ultérieur phase du processus de cycle de vie de la solution Java Enterprise System au cours de laquelle les applications distribuées sont démarrées, contrôlées, ajustées pour optimiser les performances et mises à niveau dynamiquement pour inclure une nouvelle fonctionnalité.

architecture de déploiement de référence *architecture de déploiement* qui a été conçue, implémentée et testée du point de vue de ses performances. Les architectures de déploiement de référence font office de points de départ pour la conception des architectures de déploiement des solutions personnalisées.

cas d'utilisation tâche d'utilisateur final spécifique ou ensemble de tâches effectué par une *applications d'entreprise distribuées*, puis utilisé comme base pour la conception, les tests et les mesures de performances de l'application.

Liste de références : composants Java Enterprise System

Cet appendice présente une liste de références de tous les composants Java ES, qui sont regroupés dans les catégories suivantes :

- **Description des composants de service du système.** Ces composants fournissent les services d'infrastructure clés de Java ES requis pour la prise en charge des applications d'entreprise distribuées. Ces services, décrits dans la section « [Nécessité de Java Enterprise System](#) », page 18, incluent les services de portail, les services de communication et de collaboration, les services d'identité et de sécurité, les services Web et applicatifs ainsi que les services de disponibilité.
- **Description des composants de qualité de service.** Ces composants permettent d'améliorer la qualité des services fournis par les composants de service du système ou par les composants d'applications distribuées. Parmi ces composants, on distingue les composants utilisés pour assurer un temps de disponibilité quasi continu du système, les composants d'accès permettant de prendre en charge un accès sécurisé de l'utilisateur final aux services système et les composants de gestion système destinés à améliorer l'entretien des solutions Java ES.
- **Composants partagés.** Ces composants sont des bibliothèques locales pouvant être partagées par tout composant Java ES fonctionnant sur un ordinateur hôte donné.

Dans cet appendice, les composants Java ES sont présentés par ordre alphabétique au sein de leurs catégories et sous-catégories correspondantes.

Pour obtenir une présentation de la documentation des différents composants, reportez-vous au document *Présentation de la documentation de Sun Java Enterprise System 2005T1* (<http://docs.sun.com/doc/819-0055>).

Description des composants de service du système

Les composants de service du système Java ES fournissent les services d'infrastructure requis pour la prise en charge des applications d'entreprise distribuées. Ils sont décrits dans les sections suivantes :

- [Sun Java System Access Manager 6 2005T1](#)
- [Sun Java System Application Server Enterprise Edition 8 2005T1](#)
- [Sun Java System Calendar Server 6 2005T1](#)
- [Sun Java System Directory Server 5 2005T1](#)
- [Sun Java System Instant Messaging 7 2005T1](#)
- [Sun Java System Message Queue 3 2005T1](#)
- [Sun Java System Messaging Server 6 2005T1](#)
- [Sun Java System Portal Server 6 2005T1](#)
- [Sun Java System Web Server 6.1 2005T1](#)

Sun Java System Access Manager 6 2005T1

Sun Java System Access Manager (Access Manager) fournit à une organisation une infrastructure lui permettant d'administrer les processus employés pour gérer les identités numériques des clients, des employés et des partenaires qui utilisent ses services Web et des applications non basées sur le Web. Comme ces ressources peuvent être distribuées par l'intermédiaire d'un large éventail de réseaux informatiques internes et externes, les attributs, les stratégies et les droits sont définis et appliqués à chaque identité pour gérer l'accès à ces technologies.

Le programme d'installation de Java ES fournit Access Manager sous la forme d'un composant unique installable. Si nécessaire, les sous-composants suivants d'Access Manager peuvent être installés séparément :

- **Identity Management and Policy Services Core** : permet de créer et de gérer l'identité des utilisateurs ainsi que de définir et d'évaluer les stratégies permettant d'accéder aux ressources de Java ES en fonction de l'identité des utilisateurs. Ce sous-composant comporte également les sous-composants Access Manager SDK et Delegated Administrator (voir « [Sun Java System Delegated Administrator 6 2005T1](#) », page 89).

- **Access Manager SDK** : fournit une interface distante pour Access Manager. Ce sous-composant doit être installé sur tout ordinateur qui héberge un composant Java ES ayant un accès à distance à Access Manager.
- **Console d'administration d'Access Manager** : cette interface graphique regroupe les services d'identité et la gestion de stratégie ; elle fournit aux utilisateurs une interface unique leur permettant de créer et de gérer les comptes utilisateur, les attributs de service et les règles d'accès de Directory Server.
- **Common Domain Services for Federation Management** : permet aux utilisateurs d'entrer une seule identité pour accéder aux applications proposées par plusieurs fournisseurs de services affiliés.

Sun Java System Application Server Enterprise Edition 8 2005T1

Sun Java System Application Server (Application Server) fournit une plate-forme compatible J2EE permettant le développement et le déploiement des services applicatifs et Web. Application Server fournit les services d'infrastructure permettant l'interaction entre les composants distribués à configuration groupée, notamment l'invocation de méthode distante et d'autres services de temps d'exécution.

Le programme d'installation de Java ES fournit Application Server sous la forme d'un composant unique installable. Si nécessaire, les sous-composants suivants d'Application Server peuvent être installés séparément :

- **Domain Administration Server** : fournit des fonctions d'administration côté serveur, telles que la gestion et la configuration d'Application Server, ainsi que le déploiement d'applications et de composants J2EE.
- **Application Server Administration Client** : fournit des clients d'administration graphique qui permettent de gérer et de configurer les applications hébergées et les installations d'Application Server. Administration Client aide également à déployer les applications.
- **Outil d'administration par ligne de commande** : fournit des clients d'administration par ligne de commande qui permettent de gérer et de configurer les applications hébergées et les installations d'Application Server. Cet outil aide également à déployer les applications.
- **Plug-in d'équilibrage de charge**

- **PointBase** : fournit une base de données imbriquée pouvant être utilisée pour les opérations de persistance.
- **Exemples d'applications.**

Sun Java System Calendar Server 6 2005T1

Sun Java System Calendar Server (Calendar Server) est une solution Web évolutive qui est utilisée pour la gestion de calendriers et la planification centralisées pour les entreprises et fournisseurs de services. Calendar Server prend en charge les calendriers personnels, les calendriers de groupe et les calendriers des ressources, telles que les salles de conférence et l'équipement.

Le programme d'installation de Java ES fournit Calendar Server sous la forme d'un composant unique installable.

Sun Java System Directory Server 5 2005T1

Sun Java System Directory Server (Directory Server) fournit un service d'annuaire centralisé pour les informations contenues sur votre intranet, réseau et extranet. Directory Server s'intègre aux systèmes existants et fait office de référentiel centralisé regroupant les informations relatives aux employés, aux clients, aux fournisseurs et aux partenaires. Vous pouvez étendre Directory Server pour qu'il gère les préférences et profils utilisateur et pour qu'il assure l'authentification des utilisateurs du réseau extranet.

Le programme d'installation de Java ES fournit Directory Server sous la forme d'un composant unique installable.

Sun Java System Instant Messaging 7 2005T1

Sun Java System Instant Messaging (Instant Messaging) permet aux utilisateurs finals de participer aux sessions de messagerie instantanée et de conversation, de s'envoyer des messages d'alerte et de partager des informations sur des forums instantanément. Instant Messaging est adapté aux réseaux intranet et à Internet ; il prend également en charge l'interaction avec les autres fournisseurs de messagerie instantanée.

Le programme d'installation de Java ES fournit Instant Messaging sous la forme d'un composant unique installable. Les sous-composants suivants d'Instant Messaging peuvent être installés séparément :

- **Instant Messaging Server Core** : inclut le serveur et le logiciel multiplexeur.
- **Instant Messaging Resources**.
- **Access Manager Instant Messaging Service**.

Sun Java System Message Queue 3 2005T1

Sun Java System Message Queue (Message Queue) est une solution standard au problème de communication entre les applications et de livraison fiable des messages. Message Queue est un système de messagerie d'entreprise qui implémente la norme ouverte Java Message Service (JMS).

Outre sa fonction de fournisseur JMS, Message Queue possède des fonctions qui dépassent la configuration minimale requise de la spécification JMS. Grâce au logiciel Message Queue, les processus fonctionnant sur des plates-formes et des systèmes d'exploitation différents peuvent se connecter à un service commun de Message Queue pour envoyer et recevoir des informations. Les développeurs d'applications peuvent se consacrer à la logique d'entreprise de leurs applications au lieu de devoir se charger de détails de bas niveau concernant la manière dont leurs applications communiquent sur un réseau.

Message Queue est disponible en deux éditions :

- **Enterprise Edition** (par défaut) : assure la prise en charge des services de messages multicourriers, des connexions HTTP/HTTPS, des connexions sécurisées et évolutives, du basculement de la connexion client et du support client pour le langage C. Cette édition est particulièrement bien adaptée au déploiement et au fonctionnement d'applications de messagerie dans un environnement de production à grande échelle.
- **Platform Edition** : assure une prise en charge JMS de base et convient le mieux aux environnements de développement et de déploiement à petite échelle.

Le programme d'installation de Java ES fournit Message Queue Enterprise Edition et Message Queue Platform Edition sous la forme de composants installables séparément.

Sun Java System Messaging Server 6 2005T1

Sun Java System Messaging Server (Messaging Server) est un serveur de messagerie Internet puissant et basé sur des normes qui a été conçu pour les entreprises et les fournisseurs de services. Conçu pour un traitement fiable et de haute capacité des messages, Messaging Server est constitué de plusieurs composants modulaires et configurables individuellement qui assurent la prise en charge de plusieurs protocoles de messagerie.

Le programme d'installation de Java ES fournit Messaging Server sous la forme d'un composant unique installable. Toutefois, après l'installation, chaque instance Messaging Server peut être configurée en vue de fournir n'importe lequel des nombreux services de messagerie, représentant dans le fond l'ensemble des sous-composants suivants de Messaging Server :

- **Message Store** : permet de stocker et de récupérer des messages.
- **Message Transfer Agent (MTA)** : prend en charge l'envoi de messages en traitant les connexions SMTP, en acheminant les messages et en les livrant aux stockages de messages appropriés. Peut être configuré pour une livraison des messages dans le stockage interne (entrant) ou dans le stockage externe (sortant).
- **Message Multiplexor (MMP)** : prend en charge la récupération des messages en accédant au stockage des messages (ou à un ensemble de stockages) pour les clients de messagerie utilisant les protocoles IMAP ou POP.
- **Message Express Multiplexor (MEM)** : prend en charge la récupération et l'envoi de messages par des clients de messagerie Web (HTML).

Sun Java System Portal Server 6 2005T1

Sun Java System Portal Server (Portal Server) est une solution de serveur de portail activée par identité. Portal Server associe des services de portail clés, tels que la personnalisation, l'agrégation, la sécurité, l'intégration et la recherche. Mobile access, sous-composant de Portal Server, fournit un accès sans fil à Portal Server à partir de périphériques mobiles, tels que des téléphones portables et des assistants numériques personnels (PDA).

Le programme d'installation de Java ES fournit Portal Server, Mobile access compris, sous la forme d'un composant unique installable.

Sun Java System Web Server 6.1 2005T1

Sun Java System Web Server (Web Server) est un serveur Web sécurisé à processus et à unités d'exécution multiples créé sur des normes ouvertes. Web Server assure des performances, une fiabilité, une évolutivité et une capacité de gestion excellentes pour toute entreprise, quelle que soit sa taille. Web Server prend en charge une grande diversité de normes logicielles Web, notamment JDK 1.4.1, Java Servlet 2.3, JavaServer Pages™ (JSP™) 1.2, HTTP/1.1, PKCS #11, FIPS-140, des certificats de progression de 168 bits et de nombreuses autres normes de sécurité.

Le programme d'installation de Java ES fournit Web Server sous la forme d'un composant unique installable.

Description des composants de qualité de service

Les composants utilisés pour la prise en charge des composants de service de Java ES sont regroupés dans les catégories suivantes :

- Composants de disponibilité
- Composants d'accès
- Composants administratifs

Les composants de ces catégories sont décrits dans les sections ci-après.

Description des composants de disponibilité

Les composants de disponibilité assurent un temps de disponibilité quasi continu des composants de service du système et des composants d'applications. Les composants de disponibilité suivants de Java ES sont décrits dans la section ci-dessous :

- [Sun Cluster 3.1 9/04 et Sun Cluster Agents](#)
- [High Availability Session Store 2005T1](#)

Sun Cluster 3.1 9/04 et Sun Cluster Agents

Le logiciel Sun Cluster fournit des services de disponibilité et d'évolutivité élevées pour Java Enterprise System ainsi que pour les applications basées sur l'infrastructure de Java ES.

Un cluster est un ensemble d'ordinateurs à configuration dispersée (nœuds de cluster) qui fournissent collectivement une vue client unique des services, des ressources système et des données. Au niveau interne, le cluster utilise les ordinateurs redondants, les interconnexions, le stockage de données et les interfaces réseau pour assurer une haute disponibilité aux données et services basés sur le cluster. Le logiciel Sun Cluster contrôle en permanence l'état des nœuds membres et des autres ressources du cluster et il utilise la redondance interne pour assurer un accès quasi continu à ces ressources même en cas de panne.

Le programme d'installation de Java ES fournit Sun Cluster Core et les agents Sun Cluster sous la forme de composants installables séparément. D'autres agents Sun Cluster sont disponibles sur des CD séparés.

High Availability Session Store 2005T1

Sun Java System High Availability Session Store (HADB) assure un stockage de données pouvant être utilisé pour rendre les données d'application disponibles même en cas de panne. Cette fonctionnalité est particulièrement importante pour restaurer des informations d'état associées à une session client. Sans cette fonctionnalité, une panne survenant en cours de session requiert la répétition de toutes les opérations lors du rétablissement de la session.

Les composants suivants de Java ES fournissent des services permettant de stocker des informations sur l'état de la session : Application Server, Access Manager et Message Queue. Cependant, Application Server est le seul composant pouvant utiliser les services HADB pour conserver l'état de la session pendant la panne.

Le programme d'installation de Java ES fournit HADB sous la forme d'un composant unique installable. Toutefois, il faut à la fois un serveur et un sous-composant client pour fournir des services HADB.

Description des composants d'accès

Les composants d'accès assurent un accès frontal aux services système, fréquemment depuis des sites Internet situés hors du pare-feu d'entreprise. Les composants d'accès suivants de Java ES sont décrits dans la section ci-dessous :

- [Sun Java System Communications Express 2005T1](#)
- [Sun Java System Connector pour Microsoft Outlook 6 2005T1](#)
- [Sun Java System Directory Proxy Server 5 2005T1](#)
- [Sun Java System Portal Server Secure Remote Access 6 2005T1](#)

Sun Java System Communications Express 2005T1

Sun Java System Communications Express (Communications Express) fournit un client intégré de collaboration et de communication Web qui procure aux utilisateurs finals une interface Web pour les services de calendrier, de courrier et de carnet d'adresses. Communications Express se compose de trois modules clients : calendrier, carnet d'adresses et courrier. Configurable pour assurer l'accès à Messaging Server ou à Calendar Server, voire aux deux, Communications Express fonctionne avec le schéma LDAP de Sun Java System, version 1 (schéma 1) ou schéma 2.

Le programme d'installation de Java ES fournit Communications Express sous la forme d'un composant unique installable.

Sun Java System Connector pour Microsoft Outlook 6 2005T1

Sun Java System Connector pour Microsoft Outlook permet d'utiliser Outlook comme client de bureau avec Sun Java Enterprise System. Le connecteur est un plug-in Outlook qui doit être installé sur le bureau de l'utilisateur.

Connector pour Microsoft Outlook interroge Messaging Server pour obtenir les hiérarchies de dossiers et les e-mails, puis il convertit les informations en propriétés Messaging API (MAPI) qu'Outlook peut afficher. De même, Connector utilise WCAP pour interroger Calendar Server afin d'obtenir les événements et les tâches qui sont ensuite convertis en propriétés MAPI. Avec ce modèle, Sun Java System Connector pour Microsoft Outlook crée une vue Outlook pour l'utilisateur final à partir de deux sources d'informations distinctes : courrier provenant de Messaging Server et informations de calendrier de Calendar Server.

Sun Java System Connector pour Microsoft Outlook est fourni sur le CD d'accessoires avec son propre programme d'installation.

Sun Java System Directory Proxy Server 5 2005T1

Sun Java System Directory Proxy Server (Directory Proxy Server) est un composant essentiel de tout service d'annuaire de haute importance pour les solutions d'e-commerce (commerce électronique). Directory Proxy Server est une passerelle de protocole de la couche application LDAP qui offre un contrôle amélioré de l'accès à l'annuaire, une compatibilité de schéma et une haute disponibilité à l'aide du basculement et de l'équilibrage de charge de la couche application.

Le programme d'installation de Java ES fournit Directory Proxy Server sous la forme d'un composant unique installable.

Sun Java System Portal Server Secure Remote Access 6 2005T1

Sun Java System Portal Server Secure Remote Access (Portal Server Secure Remote Access) étend Portal Server en offrant un accès distant sécurisé basé sur le navigateur au contenu et aux services de Portal Server depuis tout navigateur distant, rendant inutile l'utilisation d'un logiciel client. L'intégration à Portal Server garantit aux utilisateurs de bénéficier d'un accès sécurisé au contenu et aux services pour lesquels ils ont des droits d'accès.

Le programme d'installation de Java ES fournit Portal Server Secure Remote Access sous la forme d'un composant unique installable. Si nécessaire, les sous-composants suivants de Portal Server Secure Remote Access peuvent être installés séparément :

- **Portal Server Secure Remote Access Core.**
- **Gateway :** fournit une interface et une barrière de sécurité à un réseau intranet d'entreprise qui autorise un accès distant en dehors de l'intranet. Gateway présente à un utilisateur distant le contenu de manière sécurisée à partir de serveurs d'applications et de serveurs Web internes par le biais d'une seule interface.
- **Netlet Proxy :** permet aux utilisateurs d'exécuter en toute sécurité des services TCP/IP communs sur Internet et sur d'autres réseaux non sécurisés. Netlet permet d'exécuter des applications, telles que Telnet, SMTP, HTTP, ainsi que des applications à port fixe.
- **Rewriter Proxy :** fournit un accès sécurisé aux pages Web d'un réseau intranet d'entreprise depuis l'extérieur en transformant les liens Web et en créant des ensembles de règles permettant de traiter ces pages.

Description des composants administratifs

Les composants administratifs assurent des fonctions de gestion, telles que la configuration et le contrôle, pour les services système. Les composants administratifs suivants de Java ES sont décrits dans la section ci-dessous :

- [Sun Java System Administration Server \(et Console\) 5 2005T1](#)
- [Sun Java System Directory Preparation Tool 2005T1](#)
- [Sun Java System Delegated Administrator 6 2005T1](#)
- [Sun Remote Services Net Connect 2005T1](#)

Sun Java System Administration Server (et Console) 5 2005T1

Sun Java System Administration Server et Server Console fournissent un outil graphique permettant de gérer Directory Server, Messaging Server et Directory Proxy Server. Administration Server traite les requêtes pour les serveurs installés dans un groupe de serveurs sous le même répertoire racine, puis lance les programmes requis pour satisfaire les requêtes.

Server Console est une application Java autonome qui fonctionne avec une instance de Directory Server et une instance de Administration Server sur votre réseau. Server Console fait office d'application de gestion frontale pour le logiciel Java ES dans votre entreprise.

Le programme d'installation de Java ES fournit Server Console et Administration Server sous la forme d'un composant unique installable.

Sun Java System Directory Preparation Tool 2005T1

Sun Java System Directory Preparation Script est utilisé pour la configuration de Directory Server avec le schéma nécessaire à la création des utilisateurs pour Messaging Server et Calendar Server.

Sun Java System Delegated Administrator 6 2005T1

Sun Java System Delegated Administrator est un utilitaire de ligne de commande (`commadmin`) permettant de créer et configurer les utilisateurs, les groupes, les domaines et les ressources pour Calendar Server, Messaging Server et les autres fournisseurs de services Java ES.

Delegated Administrator est installé automatiquement lorsque vous choisissez d'installer Access Manager.

Sun Remote Services Net Connect 2005T1

Sun Remote Services Net Connect est un ensemble de services de gestion système conçus pour vous aider à mieux maîtriser votre environnement informatique. Ces services livrés depuis le Web vous permettent d'effectuer vous-même un contrôle des systèmes, de créer des rapports de performances et de tendances, et de recevoir une notification automatique des événements système de manière à pouvoir gérer plus rapidement les difficultés éventuelles avant qu'elles ne deviennent des problèmes.

Le programme d'installation de Java ES fournit Sun Remote Services Net Connect sous la forme d'un composant unique installable.

Composants partagés

Les composants partagés assurent la prise en charge de la technologie et des services locaux dont les composants de service du système et les composants de qualité de service de Java ES dépendent. Le programme d'installation de Java ES installe automatiquement les composants partagés requis pour la prise en charge des autres composants de Java ES installés sur un ordinateur hôte.

Java Enterprise System comporte les composants partagés répertoriés ci-dessous :

- Ant (Outil de compilation Jakarta ANT Java/XML)
- Apache SOAP (Simple Object Access Protocol) Runtime
- Berkeley DB
- Conteneur d'agent commun
- ICU (International Components for Unicode)
- J2SE™ (plate-forme Java 2, Standard Edition) plate-forme 5.0
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Enterprise Web Application Framework)
- JavaHelp™ Runtime
- JavaMail™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXM (Java API for XML Messaging) Client Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries) Runtime
- JAX-RPC (Java API for XML-based Remote Procedure Call) Runtime
- JCAPI (Java Calendar API)
- JDMK (Java Dynamic Management™ Kit) Runtime
- JSS (Java Security Services)
- KTSE (KT Search Engine)
- LDAP C SDK
- LDAP Java SDK
- NSPR (Netscape Portable Runtime)

- NSS (Network Security Services)
- Perl LDAP, y compris NSPERL
- SAAJ (SOAP with Attachments API for Java)
- SAML (Security Assertions Markup Language)
- SASL (Simple Authentication and Security Layer)
- SNMP (Simple Network Management Protocol) Peer
- Sun Explorer Data Collector (Solaris uniquement)
- Sun Java Monitoring Framework
- Sun Java Web Console
- Tomcat Servlet JSP Container
- XML C Library (`libxml`)
- WSCL (Web services Common Library)

A

- Access Manager
 - Composant de service du système [22](#)
 - Description [80](#)
 - Service d'infrastructure [39](#)
- Administration Server
 - Composant de qualité de service [25](#)
 - Description [89](#)
- Améliorations, *voir* Scénarios d'adoption
- Ant (Outil de compilation Jakarta ANT Java/XML) [90](#)
- Apache SOAP Runtime [90](#)
- Application Server
 - Composant de service du système [22](#)
 - Description [81](#)
 - Service d'infrastructure [39](#)
- Applications
 - Distribution, *voir* Applications d'entreprise distribuées
 - Entreprise, *voir* Applications d'entreprise distribuées
- Applications d'entreprise distribuées
 - Définition [31](#)
 - Infrastructure [18](#)
- Architecture
 - Définition [55](#)
 - Déploiement [71](#)
 - Dimensions, *voir* Dimensions architecturales
 - Introduction [33](#)
 - Logique [69](#)
 - Référence [72](#)
 - Solution [34](#)

- Architectures de déploiement
 - Conception [71](#)
 - Définition [55](#)
 - Introduction [33](#)
 - Relation avec les architectures à plusieurs niveaux [44](#)
- Architectures de déploiement de référence
 - Définition [78](#)
 - Introduction [72](#)
- Architectures logiques
 - À propos de [69](#)
 - Définition [55](#)
 - Exemple [52](#)
 - Introduction [33](#)
 - Niveaux de services d'infrastructure [36](#)
- Authentification [62](#)
- Autorisation [64](#)

B

- Berkeley DB [90](#)

C

- Calendar Server
 - Composant de service du système [22](#)
 - Description [82](#)
 - Service d'infrastructure [39](#)

- CAPI (Java Calendar API) 90
- Cas d'utilisation
 - Définition 78
 - Introduction 69
- Catégories d'utilisateurs
 - Administrateur délégué 28
 - Administrateur système 28
 - Administrateur système spécialisé 28
 - Analyste système 28
 - Architecte 28
 - Gestionnaire système 28
 - Ingénieur de projet 28
 - Intégrateur système 28
 - Planificateur d'entreprise 28
- Clients
 - Composants de service du système 21
 - Définition 55
- Clusters, *voir* Sun Cluster
- Communications Express
 - Composant de qualité de service 24
 - Description 87
- Composants
 - Accès 24
 - Administration 25
 - Définition 31
 - Dépendances 40
 - Détection des versions installées 59
 - Disponibilité 23
 - Distribution 18
 - EJB 43
 - J2EE 42
 - JSP 42
 - Partage 25
 - Partagés 90
 - Qualité de service 23, 85
 - Service système 21, 80
 - Services d'infrastructure 39
 - Servlet 42
 - Système, *voir* Composants système
- composants 18
- Composants administratifs
 - Description 88
 - Introduction 25
- Composants d'accès
 - Description 86
 - Introduction 24
- Composants d'applications
 - Architectures de niveaux logiques 42
 - Définition 55
- Composants de disponibilité
 - Description 85
 - Introduction 23
- Composants de qualité de service
 - Définition 32
 - Description 85
 - Introduction 23
- Composants de service du système
 - Définition 32
 - Dépendances 40
 - Introduction 21
- Composants EJB 43
- Composants Java Servlet 42
- Composants JSP 42
- Composants partagés 90
 - Définition 32
 - Introduction 25
- Composants système
 - Composants de qualité de service 23
 - Composants de service du système, *voir* Composants de service du système
 - Composants partagés 25, 90
 - Définition 32
 - Service système 80
- Configuration des utilisateurs 73
- Connector pour Microsoft Outlook
 - Composant de qualité de service 24
 - Description 87
- Connexion unique
 - Définition 65
 - Fonction Java ES 22, 62
 - Implémentation 63
 - Niveaux de services d'infrastructure 38
- Conteneur d'agent commun 90
- Contrôle de dépendance, programme d'installation 59

D

- Delegated Administrator
 - Composant de qualité de service 25
 - Description 89
- Dépendances 40, 59
- Déploiement
 - Architecture 71
 - Architectures de référence 72
 - Conception 71
 - Définition 77
 - Développement et personnalisation 75
 - Mise en œuvre 73
 - Mise en production 76
 - Phase du cycle de vie 70
 - Phases du cycle de vie 70
 - Scénarios, *voir* Scénarios de déploiement
 - Test de prototype 75
- Déploiement préalable
 - Définition 77
 - Phases du cycle de vie 69
- Déploiement ultérieur
 - Définition 78
 - Phases du cycle de vie 76
- Détection des logiciels installés 59
- Développement
 - Définition 77
 - Tâche de déploiement 75
- Dimensions architecturales
 - Dépendances des services d'infrastructure 36
 - Niveaux logiques 41
 - Qualité de service 45
 - Synthèse 50
- Directory Preparation Tool
 - Composant de qualité de service 25
 - Description 89
- Directory Proxy Server
 - Composant de qualité de service 24
 - Composant système 48
 - Description 87
- Directory Server
 - Composant de service du système 22
 - Description 82
 - Service d'infrastructure 39

- Disponibilité
 - Exigences 46, 48
 - Services 48, 85
- Distribution
 - Services, *voir* Services distribués
- Documentation, Java ES 12

E

- Entrée utilisateur 61
- Évolutivité
 - Exigences 46, 48
 - Services 48, 85
- Exigences d'entretien 46, 48
- Exigences de capacité latente 46
- Exigences de performances 46
- Exigences de qualité de service
 - Capacité latente 46
 - Disponibilité 46, 48
 - Entretien 46, 48
 - Évolutivité 46, 48
 - Performances 46
 - Sécurité 46, 48
- Extensions, *voir* Scénarios d'adoption

F

- Fonctions d'intégration
 - Composants partagés 20
 - Identité et sécurité 20, 60
 - Introduction 20
 - Programme d'installation intégré 20, 58
- Formation, scénarios d'adoption Java ES 30

H

- High Availability Session Store
 - Composant de qualité de service 24
 - Description 86

I

ICU (International Components for Unicode) 90

Identité

Gestion 60

Services 19, 60

Utilisateur unique 61

Identité unique

Définition 65

Introduction 61

Infrastructure

Applications d'entreprise distribuées 18

Dépendances de services, *voir* Services distribués

Instant Messaging

Composant de service du système 22

Description 82

Service d'infrastructure 39

Intégration

Fonctions, *voir* Fonctions d'intégration

Scénarios d'adoption Java ES 30

Services 38

J

J2EE

Composants 42

Modèle de composant distribué 43

Plate-forme 23

J2SE (plate-forme Java 2, Standard Edition) 90

J2SE (plate-forme Java 2, Standard Edition) 25

JAF (JavaBeans™ Activation Framework) 90

JATO (Java Studio Enterprise Web Application Framework) 90

JavaHelp 90

JavaMail 90

JAXB (Java Architecture for XML Binding) 90

JAXM (Java API for XML Messaging) 90

JAXP (Java API for XML Processing) 90

JAXR (Java API for XML Registries) 90

JAX-RPC 90

JDMK (Java Dynamic Management™ Kit) 90

JMS (Java Message Service) 22

JSS (Java Security Services) 25, 90

K

KT Search Engine (KTSE) 90

L

Langues prises en charge 58

LDAP 43, 65

LDAP C SDK 90

LDAP Java SDK 90

Linux 39

Logiciels intermédiaires 36

M

Matériel, scénarios d'adoption Java ES 30

Message Queue

Composant de service du système 22

Description 83

Service d'infrastructure 39

Messaging Server

Composant de service du système 22

Description 84

Service d'infrastructure 39

Migration, scénarios d'adoption Java ES 30

Mise en production 76

Mises à niveau, *voir* Scénarios d'adoption

N

- Niveau logique client 42
- Niveaux, logiques
 - Architecture d'application 41
 - Client 42
 - Données 43
 - Présentation 42
 - Service d'entreprise 43
- NSPR (Netscape Portable Runtime) 25, 90
- NSS (Network Security Services) 25, 91

P

- Perl LDAP 91
- Phases du cycle de vie
 - Déploiement 26, 70
 - Déploiement préalable 26, 69
 - Déploiement ultérieur 26, 76
- Plate-forme J2ME 42
- Portal Server
 - Composant de service du système 23
 - Description 84
 - Service d'infrastructure 39
- Portal Server Secure Remote Access
 - Composant de qualité de service 24
 - Composant système 48
 - Description 88
- Produits Sun Java System
 - Access Manager, *voir* Access Manager
 - Administration Server, *voir* Administration Server
 - Application Server, *voir* Application Server
 - Calendar Server, *voir* Calendar Server
 - Communications Express, *voir* Communications Express
 - Connector pour Microsoft Outlook, *voir* Connector pour Microsoft Outlook
 - Delegated Administrator, *voir* Delegated Administrator
 - Directory Preparation Tool, *voir* Directory Preparation Tool

- Directory Proxy Server, *voir* Directory Proxy Server
- Directory Server, *voir* Directory Server
- High Availability Session Store, *voir* High Availability Session Store
- Instant Messaging *voir* Instant Messaging Message Queue, *voir* Message Queue
- Messaging Server, *voir* Messaging Server
- Portal Server, Secure Remote Access, *voir* Portal Server, Secure Remote Access
- Portal Server, *voir* Portal Server
- Web Server, *voir* Web Server
- Profils utilisateur 28
- Programme de désinstallation 60
- Prototypage 75

R

- Remote Services Net Connect
 - Composant de qualité de service 25
 - Description 89
- Répertoires
 - Définition 65
 - Magasins de données utilisateur 61

S

- SAAJ (SOAP with Attachments API for Java) 91
- SAML (Security Assertions Markup Language) 91
- SASL (Simple Authentication and Security Layer) 91
- Scénarios d'adoption, Java ES
 - À propos de 29
 - Amélioration 29
 - Définition 31
 - Extension 29
 - Mise à niveau 29
 - Nouveau système 29
- Scénarios de déploiement
 - Définition 77
 - Introduction 69

Sécurité

- Exigences 46, 48
- Services 19
- Services de stratégie 38

Serveurs

- Autonomes 43
- Définition 55

Services

- Définition 32
- Entreprise 42
- Évolutivité 48, 85
- Haute disponibilité 48, 85
- Infrastructure 18
- Infrastructure, *voir* Services d'infrastructure distribués

Services administratifs 20

Services applicatifs 19, 36

Services d'accès 20

Services d'entreprise

- Définition 55
- Niveau présentation 42

Services d'exécution 37

Services de collaboration 19

Services de collaboration utilisateur 38

Services de communication 19

Services de disponibilité 20

Services de messagerie 37

Services de persistance 37

Services de plate-forme 36

Services de portail 19

Services de transport réseau 37

Services distribués

- Accès 20
- Administration 20
- Collaboration utilisateur 38
- Communication et collaboration 19
- Disponibilité 20
- Exécution 37
- Identité 19
- Infrastructure 18
- Intégration 38
- Logiciels intermédiaires 36

Messagerie 37

Niveau d'application 36

Persistance 37

Plate-forme 36, 37

Portail 19

Présentation 18

Sécurité 19, 38

Temps d'exécution 19

Transport réseau 37

Web 19

Services du système d'exploitation 37

Services système

À propos de 21

Définition 32

Services Web 19

Définition 55

SNMP (Simple Network Management Protocol) 91

SOAP (Apache) 90

Solaris 39

Solutions, Java ES

Architecture 33

Cycle de vie 26

Exemple 51

Personnalisées et prêtes à l'emploi 51

Spécifications d'implémentation 73

Stratégies

Autorisation 64

Définition 65

Sun Cluster

Agents 49

Composant de qualité de service 24

Description 85

Service de disponibilité 48

Sun Explorer Data Collector 91

Sun Java Monitoring Framework 91

Sun Java Web Console 91

Système

Composants, *voir* Composants système

Configuration 60

Services 18

T

Tâches, Java Enterprise System [26, 67](#)
Tomcat Servlet JSP Container [91](#)

U

Utilisateur, configuration [73](#)
Utilisateurs finals
 Applications distribuées [18](#)
 Définition [32](#)
Utilisateurs, *voir* Utilisateurs finals

W

Web Server
 Composant de service du système [23](#)
 Description [85](#)
 Service d'infrastructure [39](#)
WSCL (Web services Common Library) [91](#)

X

XML C Library [91](#)

