



Présentation technique de Sun Java Enterprise System 2005Q4

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

Référence : 819-3587
Octobre 2005

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué sous des licences qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et concédé sous licence par des fournisseurs de Sun.

Des parties de ce produit peuvent être dérivées des systèmes Berkeley BSD concédés sous licence par l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays et concédée exclusivement sous licence par X/Open Company, Ltd.

Sun, Sun Microsystems, le Sun logo, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou déposées de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques 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.

L'interface graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et détenteurs de licence. 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 pour l'interface graphique de Xerox couvrant également les détenteurs d'une licence Sun qui utilisent l'interface graphique OPEN LOOK et qui, en outre, se conforment aux contrats de licence écrits de Sun.

Droits soumis à la législation américaine – Logiciel commercial. Les utilisateurs de l'État sont soumis au contrat de licence standard de Sun Microsystems, Inc. ainsi qu'aux clauses applicables du FAR et de ses suppléments.

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, EST EXCLUE, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.



051220@13215



Table des matières

Préface	13
1 Présentation de Java Enterprise System	19
À quoi sert Java Enterprise System ?	19
Composants Java Enterprise System	21
Composants de service du système	22
Composants de qualité de service	24
Composants partagés	26
Utilisation de Java Enterprise System	27
Cycle de vie de la solution Java Enterprise System	27
Scénarios d'adoption de Java Enterprise System	30
Termes clés de ce chapitre	32
2 Architectures de solution Java Enterprise System	35
Structure architecturale de Java Enterprise System	35
Dimension 1 : dépendances des services d'infrastructure	37
Dimension 2 : niveaux logiques	42
Dimension 3 : qualité de service	45
Synthèse des trois dimensions architecturales	50
Exemple d'architecture de solution Java Enterprise System	50
Scénario de communication d'entreprise	51
Architecture logique de l'exemple de scénario	52
Architecture de déploiement de l'exemple de scénario	53
Termes clés de ce chapitre	53

3	Fonctions d'intégration de Java Enterprise System	55
	Programme d'installation intégré de Java Enterprise System	55
	Contrôle de logiciels préexistants	56
	Contrôle de dépendance	56
	Configuration initiale	57
	Désinstallation	57
	Services intégrés d'identité et de sécurité	57
	Identité unique	58
	Authentification et connexion unique	59
	Autorisation	61
	Termes clés de ce chapitre	61
4	Tâches du cycle de vie de la solution Java Enterprise System	63
	Tâches du cycle de vie d'une solution	63
	Déploiement préalable	65
	Déploiement	66
	Conception du déploiement	67
	Implémentation du déploiement	69
	Déploiement ultérieur	71
	Termes clés de ce chapitre	72
5	Liste de référence : composants Java Enterprise System	75
	Description des composants de service du système	76
	Sun Java System Access Manager 7 2005Q4	76
	Sun Java System Application Server Enterprise Edition 8.1 2005Q4	77
	Sun Java System Calendar Server 6 2005Q4	77
	Sun Java System Directory Server 5 2005Q4	78
	Sun Java System Instant Messaging 7 2005Q4	78
	Sun Java System Message Queue 3.6 2005Q4	78
	Sun Java System Messaging Server 6 2005Q4	79
	Sun Java System Portal Server 6 2005Q4	79
	Sun Java System Service Registry 3 2005Q4	80
	Sun Java System Web Server 6.1 2005Q4	80
	Description des composants de qualité de service	80
	Description des composants de disponibilité	81
	Description des composants d'accès	82
	Description des composants administratifs	84

Composants partagés 85

Index 87

Liste des tableaux

TABLEAU 1-1	Composants de service système Java ES	23
TABLEAU 1-2	Composants de disponibilité de Java ES	25
TABLEAU 1-3	Composants d'accès de Java ES	25
TABLEAU 1-4	Composants administratifs de Java ES	26
TABLEAU 1-5	Catégories d'utilisateur de Java ES pour les tâches du cycle de vie	29
TABLEAU 1-6	Problèmes relatifs aux scénarios d'adoption de Java ES	32
TABLEAU 2-1	Relations entre les composants de service système Java ES	41
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	47
TABLEAU 2-4	Récapitulatif des besoins d'entreprise : scénario de communications	51

Liste des figures

FIGURE 1-1	Prise en charge requise pour les applications d'entreprise distribuées 20
FIGURE 1-2	Catégories de composants Java ES 22
FIGURE 1-3	Phases du cycle de vie d'une solution et catégories d'utilisateurs 28
FIGURE 2-1	Dimensions de l'architecture de la solution Java ES 36
FIGURE 2-2	Dimension 1 : niveaux des services d'infrastructure 38
FIGURE 2-3	Composants de service système Java ES 40
FIGURE 2-4	Dimension 2 : niveaux logiques pour les applications d'entreprise distribuées 42
FIGURE 2-5	Messaging Server : exemple d'architecture à plusieurs niveaux 45
FIGURE 2-6	Conception de disponibilité à l'aide des nœuds Sun Cluster 49
FIGURE 2-7	Architecture logique d'un scénario de communication d'entreprise 52
FIGURE 3-1	Une entrée utilisateur unique prend en charge de nombreux services 59
FIGURE 3-2	Séquence d'authentification 60
FIGURE 3-3	Séquence d'autorisation 61
FIGURE 4-1	Tâches du cycle de vie d'une solution 64
FIGURE 4-2	Spécification d'un scénario de déploiement 66
FIGURE 4-3	Conversion d'un scénario de déploiement en une architecture de déploiement 67

Liste des exemples

Préface

Le *Java Enterprise System Présentation technique* aborde les concepts et informations techniques 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 de Java Enterprise System.

Public concerné par ce manuel

Le *Java Enterprise System Présentation technique* s'adresse aux personnes chargées de la conception, du déploiement ou de la maintenance de 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 lecteurs du *Java Enterprise System Présentation technique* doivent posséder une certaine maîtrise des 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;

Documentation Java Enterprise System

L'ensemble de documents relatifs à Java ES décrit la planification du déploiement et l'installation système. L'URL de la documentation système est <http://docs.sun.com/coll/1286.1>. Pour une introduction à Java ES, consultez les manuels dans l'ordre où ils apparaissent dans le tableau suivant.

TABLEAU P-1 Documentation Java Enterprise System

Titre du document	Contenu
<i>Notes de version de Sun Java Enterprise System 2005Q4</i>	Contient les informations les plus récentes concernant Java ES, 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 2005Q4</i>	Fournit la description de toute la documentation relative à Java ES, en tant que système et pour chaque composant.
<i>Présentation technique de Sun Java Enterprise System 2005Q4</i>	Décrit les principes techniques et conceptuels de Java ES. Décrit les composants, l'architecture, les processus et les fonctions.
<i>Guide de planification du déploiement de Sun Java Enterprise System 2005Q4</i>	Présente les modalités de planification et de conception des solutions de déploiement d'entreprise basées sur Java ES. Fournit les 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 ES.
<i>Guide de planification de l'installation de Java ES System 2005Q4</i>	Vous aide à développer les spécifications d'implémentation du matériel, du système d'exploitation et des aspects réseau du déploiement de Java ES. Décrit les problèmes, comme les dépendances entre composants, à résoudre dans votre planification d'installation et de configuration.
<i>Guide d'installation de Sun Java Enterprise System 2005Q4 pour UNIX</i>	Vous guide tout au long de la procédure d'installation de Java ES sous Solaris ou Linux. Il vous explique comment configurer les composants après leur installation et comment vous assurer de leur fonctionnement correct.

TABLEAU P-1 Documentation Java Enterprise System (Suite)

Titre du document	Contenu
<i>Référence de l'installation de Java ES System 2005Q4</i>	Fournit des informations supplémentaires sur les paramètres de configuration, contient des fiches d'information à utiliser pour la planification de la configuration, et répertorie les références, telles que les répertoires par défaut et les numéros de ports.
<i>Sun Java Enterprise System 2005Q1 Deployment Example Series: Evaluation Scenario</i>	Décrit la procédure d'installation de Java ES sur un système, de mise en place des principaux services réseau partagés et de configuration des comptes utilisateur pouvant accéder à ces services.
<i>Guide de mise [00e0] niveau de Sun Java Enterprise System 2005Q4</i>	Fournit les instructions de mise à niveau de Java ES sous Solaris et Linux.
<i>Sun Java Enterprise System Glossary</i>	Définit les termes utilisés dans la documentation Java ES.

Conventions typographiques

Le tableau suivant répertorie les modifications typographiques utilisées dans ce manuel.

TABLEAU P-2 Conventions typographiques

Type de caractères	Signification	Exemple
<i>AaBbCc123</i>	Nom des commandes, fichiers et répertoires, et messages système.	Modifiez votre fichier <code>.login</code> . Pour établir la liste de tous les fichiers, entrez <code>ls -a</code> . <code>machine_name%</code> vous avez un message.
AaBbCc123	Ce que vous entrez, et non ce qui apparaît sur l'écran de l'ordinateur.	<code>machine_name%</code> su Password:
<i>AaBbCc123</i>	Substituant à remplacer par un nom réel ou une valeur.	La commande de suppression d'un fichier est <code>rm filename</code> .

TABLEAU P-2 Conventions typographiques (Suite)

Type de caractères	Signification	Exemple
<i>AaBbCc123</i>	Titres de manuels, nouveaux termes et termes à mettre en valeur (certains éléments mis en valeur sont affichés en gras)	Lisez le chapitre 6 du <i>Guide de l'utilisateur</i> . Un <i>cache</i> est une copie enregistrée localement. N'enregistrez <i>pas</i> le fichier.

Accès aux ressources Sun en ligne

Le site Web docs.sun.comSM vous permet d'accéder à la documentation technique Sun en ligne. Vous pouvez parcourir les archives docs.sun.com ou rechercher un titre de manuel ou un sujet spécifique. Les manuels sont disponibles en ligne aux formats PDF et HTML. Ces deux formats sont compatibles avec les aides technologiques proposées aux utilisateurs souffrant d'un handicap physique.

Pour accéder aux ressources Sun suivantes, visitez le site <http://www.sun.com> :

- Téléchargement de produits Sun
- Services et solutions
- Support (y compris patchs et mises à jour)
- Formation
- Recherche
- Communautés (par exemple, réseau de développeurs Sun)

Références aux sites Web tiers

Des adresses URL de sites tiers, qui renvoient à des informations complémentaires connexes, sont référencées dans ce document.

Remarque – Sun décline toute responsabilité quant à la disponibilité des sites Web tiers mentionnés. Sun ne garantit pas le contenu, la publicité, les produits et autres matériaux disponibles sur ces sites ou dans ces ressources, ou accessibles par leur intermédiaire, et ne saurait en être tenu pour responsable. Par ailleurs, la responsabilité de Sun ne saurait être engagée en cas de dommages ou de pertes, réels ou supposés, occasionnés par, ou liés à, l' utilisation du contenu, des produits ou des services disponibles sur ces sites ou dans ces ressources, ou accessibles par leur biais, ou encore à la confiance qui a pu leur être accordée.

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 des commentaires. Dans le formulaire en ligne, indiquez le titre et le numéro du document. Ce numéro est constitué de sept ou neuf chiffres. Il figure sur la page de titre du manuel et dans l'URL du document. Par exemple, le numéro de ce manuel est 819-3587.

Présentation de Java Enterprise System

Sun Java™ Enterprise System (Java ES) est un ensemble de composants logiciels fournissant les services nécessaires pour prendre en charge les applications essentielles de l'entreprise, distribuées sur un réseau ou un environnement 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 qu'implique l'utilisation du système. Il couvre les rubriques suivantes :

- "À quoi sert Java Enterprise System ?" à la page 19
- "Composants Java Enterprise System " à la page 21
- "Utilisation de Java Enterprise System" à la page 27
- "Termes clés de ce chapitre" à la page 32

À quoi sert Java Enterprise System ?

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

Java Enterprise System fournit les services d'infrastructure nécessaires pour prendre en charge des [applications d'entreprise distribuées](#), c'est-à-dire des applications qui ont généralement les caractéristiques suivantes :

- **Distribuées** : l'application se compose de [composants](#) logiciels interactifs déployés dans un environnement réseau pouvant inclure des sites géographiquement distants. Ces composants distribués, exécutés sur les différents ordinateurs de

l'environnement, fonctionnent conjointement pour fournir des fonctions d'exploitation spécifiques aux **utilisateurs finals** et aux autres applications d'entreprise.

- **Niveau entreprise** : la portée et l'étendue de l'application répond aux besoins d'un environnement de production ou d'un fournisseur de service 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 nécessitent un ensemble de **services** d'infrastructure sous-jacents permettant aux composants distribués de communiquer les uns avec les autres, de coordonner leur travail, d'implémenter un accès sécurisé, etc. Ces services d'infrastructure sont, à leur tour, pris en charge par un environnement matériel constitué 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 l'essentiel, Java Enterprise System fournit le niveau services d'infrastructure distribuée présentée dans la **Figure 1-1**. Cependant, les services Java Enterprise System comportent également plusieurs services de niveau application qui sont accessibles aux utilisateurs finals.

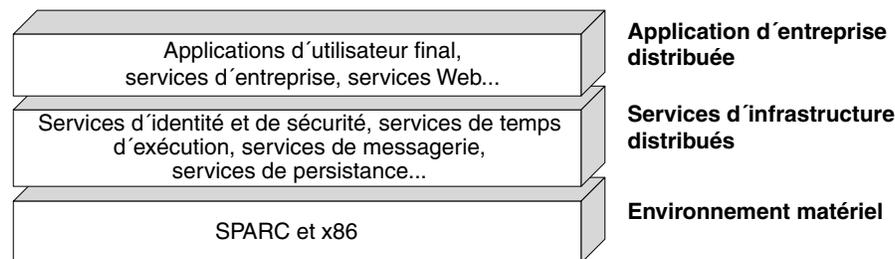


FIGURE 1-1 Prise en charge requise pour les applications d'entreprise distribuées

Parmi les services fournis par Java Enterprise System, on peut citer les services suivants :

- **Services de portail** : Ces services permettent aux employés, télétravailleurs, experts, partenaires commerciaux, fournisseurs et clients d'accéder aux ressources de l'entreprise quelle que soit leur emplacement, à l'intérieur ou à l'extérieur du réseau de l'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 communications 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 (messagerie instantanée, conférences) ainsi que la planification du travail de l'utilisateur.

- **Services de sécurité et d'identité réseau** : Ces services améliorent la sécurité et la protection des informations clés de l'entreprise en garantissant que les stratégies de contrôle d'accès appropriées soient appliquées à toutes les communautés, applications et services sur une base globale. Ils 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 les uns avec les autres lors de l'exécution et prennent en charge le développement, le déploiement et la gestion d'applications pour une large gamme de serveurs, clients et 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'application et pour 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 de Java Enterprise System.
- **Services administratifs** : Ces services aident à préserver et à régler les performances des applications prises en charge par Java Enterprise System.

Vous pouvez déployer un ou plusieurs services Java Enterprise System, chacun d'eux pouvant inclure un certain nombre de composants Java Enterprise System.

Composants Java Enterprise System

Java Enterprise System est une intégration de composants et produits logiciels distincts sous la forme d'un système logiciel unique. Les composants de ce système ont été testés ensemble pour vérifier leur interopérabilité. Leur intégration est facilitée par un certain nombre de fonctions au niveau du système :

- Tous les composants sont synchronisés sur un ensemble commun de bibliothèque partagées.
- Tous les composants Java ES sont installés à l'aide d'un programme d'installation unique.
- Tous les composants Java ES peuvent partager une identité utilisateur et un système de gestion de la sécurité intégrés.

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 indiqué dans l'illustration suivante :

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

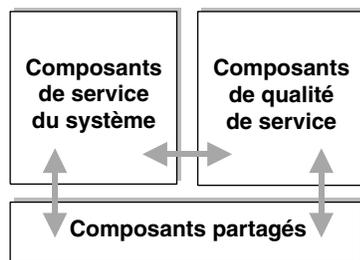


FIGURE 1-2 Catégories de composants Java ES

Composants de service du système

Plusieurs composants Java ES fournissent les principaux services qui prennent en charge les solutions logicielles distribuées. Ces **services système** comprennent les services de portail, les services de communication et de collaboration, les services d'identité et de sécurité, les services de conteneur Web, ainsi que les services d'application J2EE.

Les **composants de service système** qui fournissent ces services distribués, et les services qu'ils fournissent sont décrits brièvement dans le tableau suivant. Chaque composant de service système est un processus serveur multi-thread capable de prendre en charge un nombre important de clients. Pour plus de détails sur chaque composant, reportez-vous à la section "[Description des composants de service du système](#)" à la page 76

TABLEAU 1-1 Composants de service 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 des accès englobent l'authentification (notamment la connexion unique) et l'autorisation basée sur les rôles pour l'accès aux applications et/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 des beans de session, des beans d'entité et des beans pilotés par messages. 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 comprend un client basé sur un navigateur qui agit en interaction 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. Les services comprennent un gestionnaire de présence indiquant aux utilisateurs qui est actuellement en ligne, ainsi qu'un client de type 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 API Java Message Service (JMS) et ajoute des fonctions d'entreprise telles que la sécurité, l'évolutivité et l'administration à distance.

TABLEAU 1-1 Composants de service système Java ES (Suite)

Composant	Services système fournis
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 bases de messages et offre des services de filtrage de contenu permettant de rejeter automatiquement les courriers indésirables et de bloquer les attaques virales.
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 composant Web Java tels que les composants Java Servlet et JavaServer Pages™ (JSP™). Web Server prend également en charge d'autres technologies d'application Web pour la fourniture de contenus Web statiques et dynamiques, tels que les scripts CGI et Active Server Pages.
Sun Java System Service Registry	Fournit un registre et un référentiel pour la prise en charge des applications SOA (Service-Oriented Architecture). Service Registry implémente des normes industrielles pour l'enregistrement et la détection des services Web, ainsi que pour la gestion des informations et faits associés, des artefacts, par exemple XMLSchema, des règles des processus industriels, du contrôle d'accès, du contrôle de version, etc.

Composants de qualité de service

Outre les composants de service système présentés dans le [Tableau 1-1](#), Java Enterprise System comprend un certain nombre de composants utilisés pour améliorer la qualité des services offerts par les composants de service système. Les composants de qualité de service peuvent également améliorer les services d'application dont le développement a été personnalisé. Les [composants de qualité de service](#) se répartissent 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 services d'applications personnalisés. Le tableau suivant présente les composants de disponibilité inclus dans Java Enterprise System, ainsi que les services qu'ils fournissent. Pour plus de détails sur chaque composant, reportez-vous à la section "[Description des composants de disponibilité](#)" à la page 81.

TABLEAU 1-2 Composants de disponibilité de 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 et sécurisé aux services système, la plupart du temps à partir de sites Internet situés hors du pare-feu d'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 plus de détails sur chaque composant, reportez-vous à la section "[Description des composants d'accès](#)" à la page 82.

TABLEAU 1-3 Composants d'accès de Java ES

Composant	Services d'accès fournis
Sun Java System Directory Proxy Server	Fournit des services de sécurité pour Directory Server à partir de l'extérieur d'un pare-feu d'entreprise. Directory Proxy Server assure le contrôle d'accès à l'annuaire et le routage des instances multiples de Directory Server.
Sun Java System Web Proxy Server	Assure la mise en cache, le filtrage et la distribution du contenu Web pour les requêtes Internet entrantes et sortantes.
Sun Java System Communications Express	Fournit un accès Web à Messaging Server, Calendar Server et Directory Server, selon la configuration.
Sun Java System Portal Server Secure Remote Access	Fournit un accès Internet sécurisé à partir de l'extérieur d'un pare-feu d'entreprise aux contenus et services de Portal Server, notamment les portails internes.
Sun Java System Connector for Microsoft Outlook	Fournit aux clients de bureau utilisant Microsoft Outlook une interface vers 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 plus de détails sur chaque composant, reportez-vous à la section "Description des composants administratifs" à la page 84.

TABLEAU 1-4 Composants administratifs de Java ES

Composant	Services administratifs fournis
Sun Java System Administration Server (et Console)	Fournit un outil d'administration graphique vous permettant de configurer et de gérer Directory Server et Messaging Server.
Sun Java System Directory Preparation Tool	Fournit un script pour la configuration de Directory Server, avec le schéma nécessaire pour le provisioning des utilisateurs de Messaging Server et Calendar Server.
Sun Java System Delegated Administrator	Fournit des outils de ligne de command et d'interface graphique pour insérer les entrées utilisateur dans Directory Server avec les attributs utilisateur requis par Messaging Server et Calendar Server.

Composants partagés

Java Enterprise System comprend un certain nombre de bibliothèques partagées installées localement, dont dépendent de nombreux composants de service système et de qualité de service. Les [composants partagés](#) Java ES fournissent des services locaux aux composants Java ES exécutés sur le même ordinateur hôte.

Les composants partagés sont fréquemment utilisés pour assurer la portabilité sur différents systèmes d'exploitation. Exemples de composants partagés de Java Enterprise System : Java 2 Platform, Standard Edition (plate-forme J2SE™), Netscape Portable Runtime (NSPR), Network Security Services (NSS), Network Security Services for Java (JSS), etc.. Pour la liste complète, reportez-vous à la section "Composants partagés" à la page 85.

Les composants partagés sont installés automatiquement par le programme d'installation de Java ES, en fonction des composants de service système et de qualité de service qui sont installés.

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 complexité de ces tâches varient selon le point de départ de l'adoption de Java Enterprise System et la nature de la solution que vous tentez 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 intervenant dans la création de solutions d'entreprise basées sur le logiciel Java ES peuvent être divisées en plusieurs phases, comme illustré dans la [Figure 1-3](#). Cette figure présente également la catégorie d'utilisateurs de Java Enterprise System qui effectue les diverses tâches en règle générale.

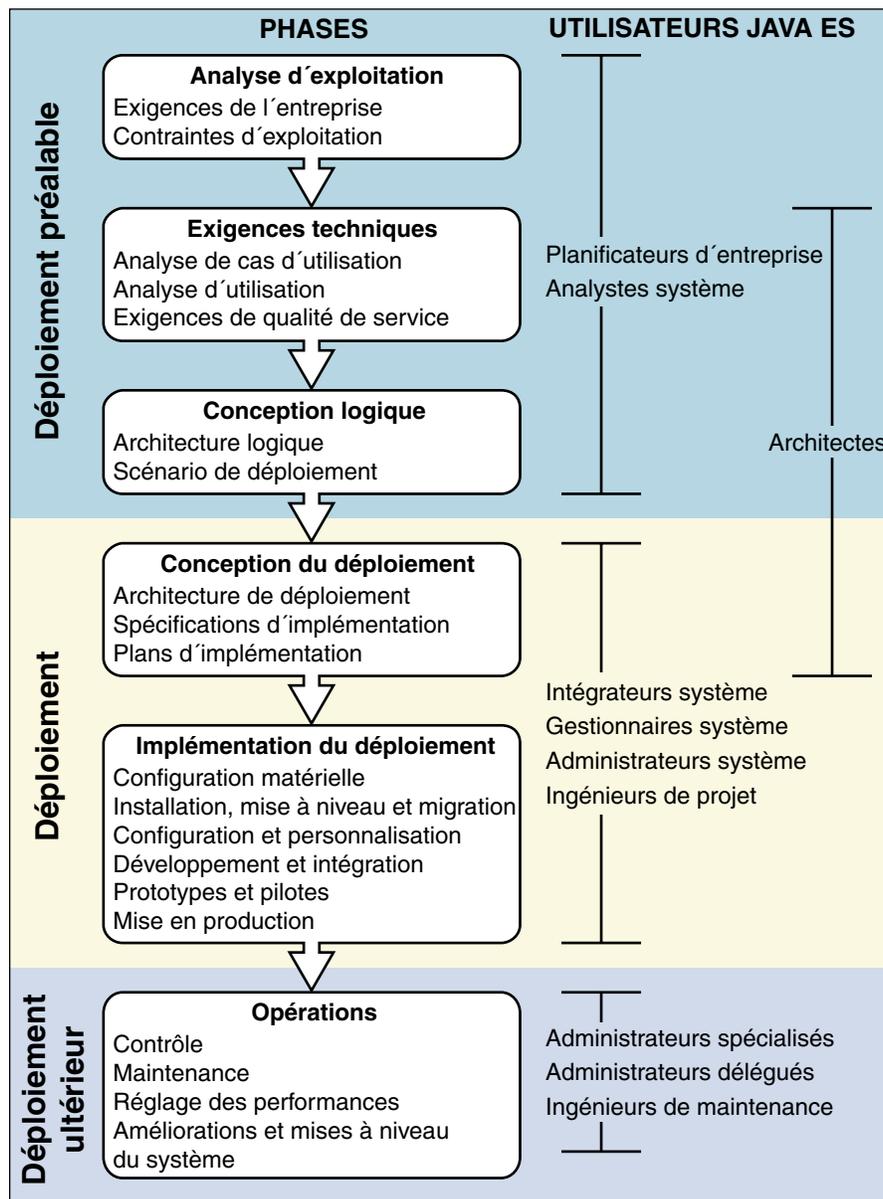


FIGURE 1-3 Phases du cycle de vie d'une solution et catégories d'utilisateurs

Les phases du cycle de vie indiquées dans la Figure 1-3 peuvent être divisées en trois grands groupes :

- **Déploiement préalable** : au cours des phases de cette partie du cycle de vie, les besoins d'une entreprise sont convertis en un scénario de déploiement, c'est-à-dire en une architecture logique et en 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'exécution, une solution déployée est exécutée dans des conditions de production et ses performances sont surveillées et optimisées. La solution déployée est également mise à niveau afin d'inclure au besoin les nouvelles fonctionnalités.

Le cycle de vie de la solution et les tâches de chacune des phases illustrées dans la [Figure 1-3](#) sont étudiées plus en détail dans le [Chapitre 4](#).

La [Figure 1-3](#) présente les utilisateurs de Java ES qui réalisent généralement les tâches des différentes phases du cycle de vie. Si vous utilisez Java ES, vous devez vous situer dans l'une ou l'autre des catégories d'utilisateurs présentées dans la [Figure 1-3](#). Le tableau suivant décrit les connaissances et l'expérience correspondant à chaque catégorie d'utilisateurs.

TABLEAU 1-5 Catégories d'utilisateur de Java ES pour les tâches du cycle de vie

Utilisateur	Connaissances et expérience	Phases
Planificateur Analyste système	Possède des connaissances générales et non des connaissances techniques approfondies. Sait en quoi consiste la direction stratégique d'une entreprise. Connaît les processus, les objectifs et les exigences d'entreprise.	Analyse d'exploitation 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

TABLEAU 1-5 Catégories d'utilisateur de Java ES pour les tâches du cycle de vie (Suite)

Utilisateur	Connaissances et expérience	Phases
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 ES peuvent varier considérablement. Toutefois, l'objectif principal de tout déploiement Java ES s'inscrit dans l'un des scénarios d'adoption suivants :

- **Nouveau système** : Si ne disposez encore d'aucun système logiciel, vous pouvez déployer le logiciel Java Enterprise System afin de prendre en charge une nouvelle solution d'entreprise.
- **Amélioration** : Si vous disposez déjà d'une infrastructure informatique, vous allez remplacer un, plusieurs ou la totalité des éléments par le logiciel Java ES. La plupart du temps, 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** : Si vous disposez déjà d'une infrastructure informatique, vous pouvez déployer le logiciel Java Enterprise System ne faisant pas actuellement partie de votre système. 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 ES ou l'autorisation et l'authentification Java pour les services existants.

- **Mise à niveau** : En partant d'une infrastructure informatique composée d'une version antérieure de Java Enterprise System ou de produits Sun antérieurs à Java Enterprise System, vous effectuez une mise à niveau pour obtenir la dernière version des composants Java Enterprise System.

Chaque scénario d'adoption contient ses propres problèmes et défis. Quel que soit le scénario d'adoption correspondant à votre situation, le cycle de vie de la solution illustrée dans 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 d'une infrastructure existante à l'aide d'un nouveau logiciel requiert souvent la migration de données du système existant vers le nouveau. Ces données peuvent correspondre à des informations de configuration, des informations utilisateur ou des informations d'applications. En outre, vous devrez peut-être migrer la logique d'entreprise ou de présentation en raison des nouvelles interfaces de programmation.
- **Intégration** : L'ajout d'un nouveau logiciel à un système existant ou le remplacement de sous-systèmes logiciels requiert souvent l'intégration des 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** : Une modification de l'infrastructure implique presque systématiquement des changements dans les procédures et compétences 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, des contraintes économiques peuvent vous amener à réutiliser 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 ES.

TABLEAU 1-6 Problèmes relatifs aux scénarios d'adoption de 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. ¹
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
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

¹ 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 Java Enterprise System.

- scénario d'adoption** Raison principale justifiant le déploiement du logiciel Java Enterprise System et caractérisant le système logiciel de départ et l'objectif que vous cherchez à atteindre. Il existe principalement quatre scénarios d'adoption de Java Enterprise System : nouveau système, remplacement, extension et mise à niveau.
- composant** Unité de logique logicielle à partir de laquelle sont élaborées les applications distribuées. Un composant peut-être l'un des **composants système** inclus dans Java Enterprise System ou un **composant d'application** au développement 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, seuls ou combinés, fournissent des **services d'entreprise** et peuvent être encapsulés sous la forme de **services Web**.

application d'entreprise distribuée	Application dont la logique s'étend sur un réseau ou un environnement Internet (aspect distribué) et dont la portée ou l'échelle répond aux besoins d'un environnement de production ou d'un fournisseur de services (aspect entreprise).
utilisateur final	Personne utilisant une application distribuée, souvent par le biais d'une interface utilisateur graphique, telle qu'un navigateur Internet ou une interface de périphérique mobile. Le nombre d'utilisateurs pris en charge simultanément 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 être exécutée à un niveau très bas, comme la gestion de la mémoire, ou à un niveau très élevé, comme pour un service d'entreprise de contrôle 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	Une des nombreuses sortes de composants système inclus dans Java Enterprise System. Les 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.
composant partagé	Une des nombreuses sortes 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. Par opposition, un composant de service du système fournit des services d'infrastructure distribués aux autres composants du système (ou aux composants d'application).
composant système	Tout package logiciel 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 service système en fournissant les services d'accès et d'administration 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 offerte par Java Enterprise System. Les services système exigent normalement la prise en charge d'un certain nombre de composants de qualité de service, de composants partagés ou les deux.
composant de service du système	Une des nombreuses sortes de composants système inclus dans Java Enterprise System. Les composants de services du système fournissent les principaux services d'infrastructure de Java Enterprise System : services de portail, de communication et de collaboration, d'identité et de sécurité, Web et applicatifs, et de disponibilité.

Architectures de solution Java Enterprise System

Ce chapitre fournit un aperçu des concepts architecturaux sur lesquels sont basés 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 solution Java ES ont 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 Java ES jouent un rôle important à la fois dans les architectures logiques et les architectures 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.

Il aborde les sujets suivants :

- ["Structure architecturale de Java Enterprise System"](#) à la page 35
- ["Exemple d'architecture de solution Java Enterprise System"](#) à la page 50
- ["Termes clés de ce chapitre"](#) à la page 53

Structure architecturale de Java Enterprise System

Les composants de Java ES 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 comprend les trois dimensions architecturales suivantes :

- **Dépendances des services d'infrastructure** : Cette dimension met l'accent sur le rôle des composants de service du système (voir "[Composants de service du système](#)" à la page 22) dans les solutions distribuées de support.
- **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](#)" à la page 24).

Ces trois dimensions sont illustrées dans la figure suivante.

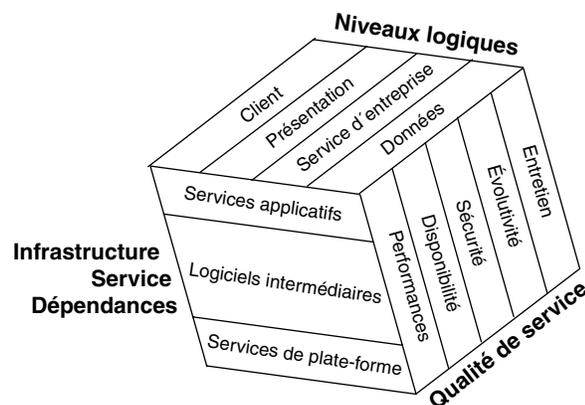


FIGURE 2-1 Dimensions de l'architecture de la solution Java ES

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 communicants des applications d'entreprise distribuées nécessitent un jeu de services d'infrastructure sous-jacent, permettant aux composants distribués de communiquer entre eux, de coordonner leur travail, d'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

Lorsque vous concevez un système logiciel distribué, qu'il s'agisse essentiellement de composants au développement personnalisé ou de composants Java ES standard, vous devez incorporer un certain nombre de services d'infrastructure. Ces services fonctionnent sur plusieurs niveaux.

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

La hiérarchie des services de la [Figure 2-2](#) et les dépendances entre eux 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 "Composants de service du système" à la page 22).

En général, les services présentés dans la [Figure 2-2](#) se répartissent en trois grands groupes : les services de plate-forme du niveau inférieur, les services d'application du niveau supérieur et un groupe de services intermédiaires, ainsi nommés d'après leur emplacement entre les deux autres groupes.

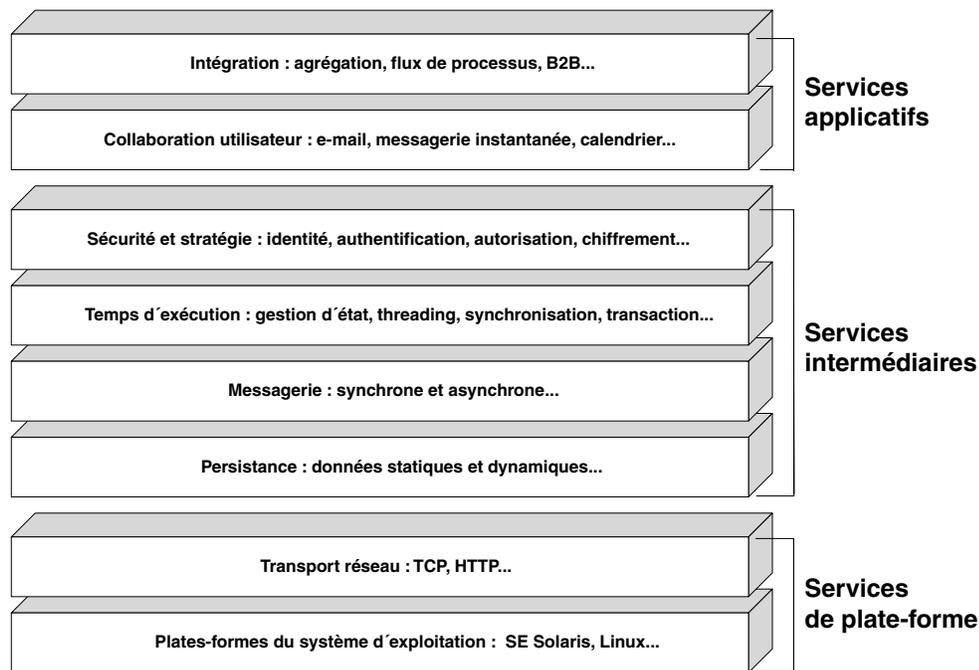


FIGURE 2-2 Dimension 1 : niveaux des services d'infrastructure

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 service sont décrits du niveau le plus bas au niveau le plus élevé, comme illustré dans la Figure 2-2:

- **Plates-formes de système d'exploitation** : assure la prise en charge de base de tout processus exécuté sur l'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 la prise en charge réseau de base pour les communications entre les composants d'application distribués exécutés sur des ordinateurs différents. 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.
- **Persistence** : assure la prise en charge pour les accès et le stockage des données statiques (informations sur l'utilisateur, le répertoire ou la configuration) et des 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 des accès sécurisés aux ressources d'application. Ces services incluent la prise en charge des stratégies régissant les accès des groupes ou par rôles aux ressources distribuées, ainsi que les possibilités 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 les services qui regroupent les services d'exploitation 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 service présentés dans la [Figure 2-2](#) reflètent une dépendance générale de différents services d'infrastructure les uns par rapport aux autres, des services de système d'exploitation de bas niveau aux services d'intégration et d'application de plus haut niveau. 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 représentent pas une couche stricte de 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 exécutables peuvent dépendre directement de services de plate-forme, sans nécessiter le moindre niveau de service intermédiaire. De plus, d'autres niveaux de services, tels que le contrôle ou le service de gestion, peuvent être inclus dans cette illustration conceptuelle.

Composants de service d'infrastructure 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 système Java ES au sein des différents niveaux est illustré dans la Figure 2-3.

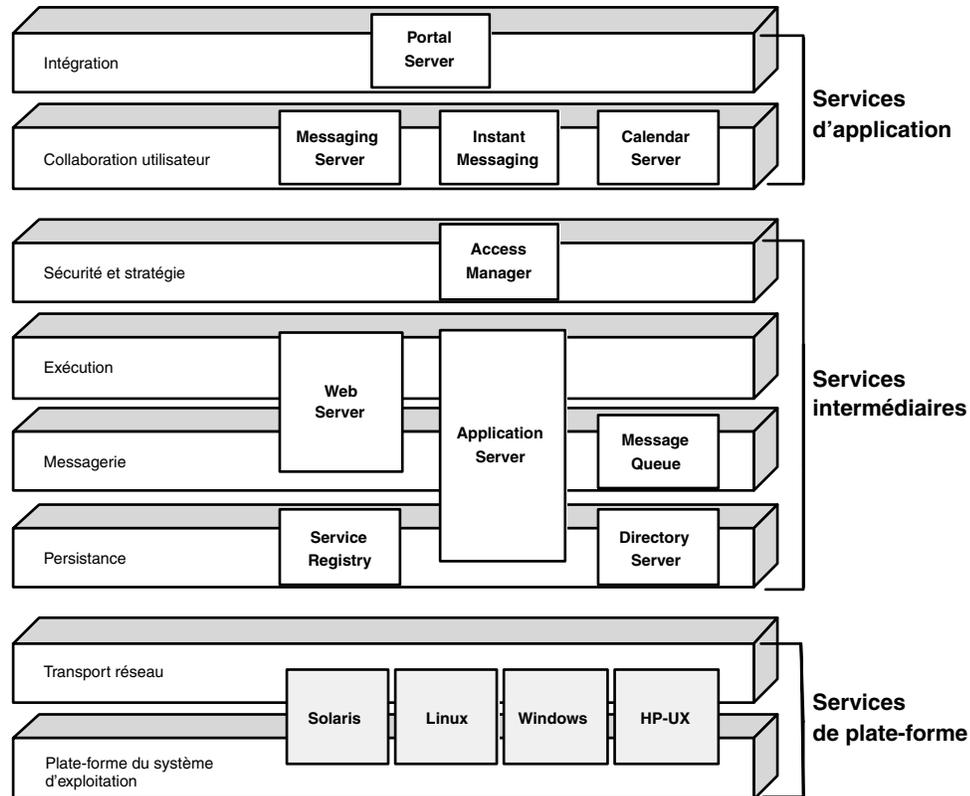


FIGURE 2-3 Composants de service système Java ES

Remarque – Les plates-formes de système d'exploitation illustrées dans la Figure 2-3 ne font pas véritablement partie de Java Enterprise System ; toutefois, elles ont été incluses afin de montrer les plates-formes de système d'exploitation sur lesquelles les composants Java ES sont pris en charge.

Dépendances de service d'infrastructure Java Enterprise System

En général, chaque composant de service système Java ES présenté dans la [Figure 2-3](#) dépend des composants situés au-dessous de lui dans l'infrastructure et prend en charge les composants se trouvant 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](#) montre les relations spécifiques entre les composants de service système Java ES, en partant du haut vers le bas, comme illustré dans la [Figure 2-3](#).

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

Composant	Dépend de	Prend en charge
Portal Server	Application Server ou Web Server Access Manager Directory Server Si 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 Si configuré pour une connexion unique : Calendar Server Messaging Server Instant Messaging
Application Server	Message Queue Directory Server (pour les objets gérés)	Portal Server Access Manager
Message Queue	Directory Server (pour les objets gérés)	Application Server

TABLEAU 2-1 Relations entre les composants de service système Java ES (Suite)

Composant	Dépend de	Prend en charge
Web Server	Access Manager (pour le contrôle d'accès)	Portal Server Access Manager
Directory Server	aucune.	Portal Server Calendar Server Messaging Server Instant Messaging Access Manager
Registre de service	Aucune dépendance	Composants basés sur Application Server

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 dans la figure suivante.

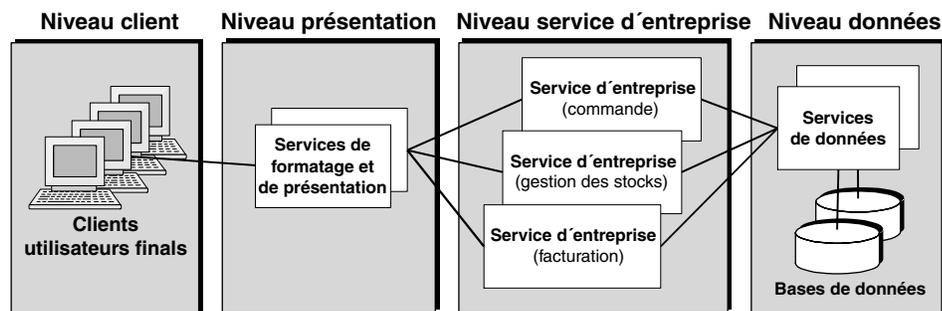


FIGURE 2-4 Dimension 2 : niveaux logiques pour les applications d'entreprise distribuées

Pour l'essentiel, les architectures de niveau logique correspondant à la couche d'application d'entreprise distribuée de la [Figure 1-1](#) . Les composants de service système Java ES étudiés dans la section "Niveaux de services d'infrastructure" à la [page 37](#) assurent la prise en charge des composants d'application dans tous les niveaux logiques illustrés dans la [Figure 2-4](#). Toutefois, les concepts de niveau logique s'appliquent également aux composants de service système assurant des services de niveau application, comme Messaging Server et Calendar Server.

Description des niveaux logiques

Cette section fournit une brève description des quatre niveaux logiques représentés dans 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 comprend la logique d'application qui prépare les données en vue de leur livraison au niveau client et traite les requêtes émanant du niveau client en vue de les livrer à la logique d'entreprise d'arrière-plan. La logique du niveau présentation comprend généralement des composants J2EE tels que les composants Java Servlet ou les composants JSP qui préparent les données en vue de leur livraison en format HTML ou XML ou qui reçoivent les requêtes de traitement. Ce niveau doit également inclure un service de portail pouvant fournir un accès personnalisé et sûr aux [services d'entreprise](#) dans le niveau service d'entreprise.
- **Niveau service d'entreprise** : le niveau service d'entreprise comprend la logique qui exécute 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 général, ce niveau se compose de composants étroitement associés conformes au modèle de composant distribué J2EE, par exemple, les objets Java, les composants EJB ou les beans gérés par messages. 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 assemblages de services peuvent être encapsulés sous la forme de [services Web](#) associés au sein d'un modèle d'architecture orienté service et respectant les normes d'interface SOAP (Simple Object Access Protocol, protocole SOAP). Les services d'entreprise peuvent également être élaborés sous la forme de [serveurs](#) autonomes, tels qu'un serveur de calendrier d'entreprise ou un serveur de messagerie.
- **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 dans la [Figure 2-4](#) met en évidence l'indépendance logique et physique des composants, représentés par quatre niveaux distincts. Ces niveaux reflètent le partitionnement de la logique d'application sur les divers ordinateurs d'un 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 et sur plusieurs types de plates-forme matérielle (c'est-à-dire différentes configurations de processeur, différents chipset 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 méthode de mappage des composants d'application ou d'infrastructure avec un environnement matériel (c'est-à-dire votre architecture de déploiement) dépend de plusieurs facteurs, notamment 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 comme 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 indiqué dans la [Figure 2-3](#), les composants de service d'infrastructure de Java ES fournissent l'infrastructure de support 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 d'un certain nombre de configurations logiques distinctes de Messaging Server. Ces configurations distinctes offrent chacune un ensemble de services distinct. 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.

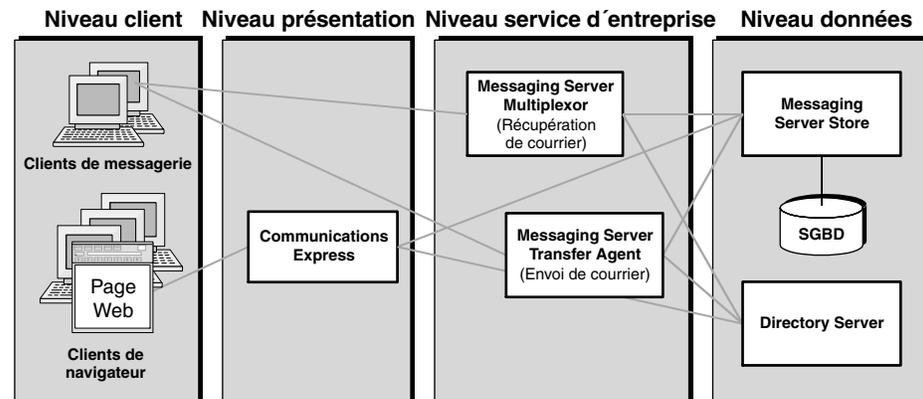


FIGURE 2-5 Messaging Server : exemple d'architecture à plusieurs niveaux

Remarque – La Figure 2-5 ne représente pas une architecture logique complète ; un certain nombre de composants Java ES ont été omis afin de simplifier le schéma. 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 offre la souplesse nécessaire pour répondre aux exigences de qualité de service (voir "Dimension 3 : qualité de service" à la page 45). 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) concernent surtout les aspects logiques de l'architecture, notamment les composants et interactions requis pour fournir des services aux utilisateurs finals. Toutefois, la capacité d'une solution à satisfaire aux exigences de qualité de service constitue une dimension tout aussi importante en matière de déploiement.

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 de l'importance prise par les services Web et d'e-commerce dans les opérations des entreprises, 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 (temps d'activité 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.
Facilité de maintenance	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 a un impact considérable sur 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 étroitement 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 les performances qui, à leur tour, sont susceptibles 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 les équilibrateurs de charge et les pare-feux.

Les composants de qualité de service de Java ES, présentés dans la section “Composants de qualité de service” à la page 24, sont récapitulés ci-après :

- **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

TABLEAU 2-3 Composants de qualité de service et qualités système influencées (Suite)

Composant	Qualités système influencées
Sun Cluster	Disponibilité Évolutivité
Web Proxy Server	Sécurité Performances Facilité de maintenance

Logiciel Sun Cluster

Le logiciel Sun Cluster offre des services haute disponibilité Évolutivité pour les composants Java ES et pour les applications prises en charge par l'infrastructure Java ES.

Un cluster est un ensemble d'ordinateurs interconnectés fournissant conjointement une vue client unique des services, ressources du système et 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. 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 représente un cluster à deux nœuds prenant en charge les services de magasins de données pour Messaging Server et Calendar Server.

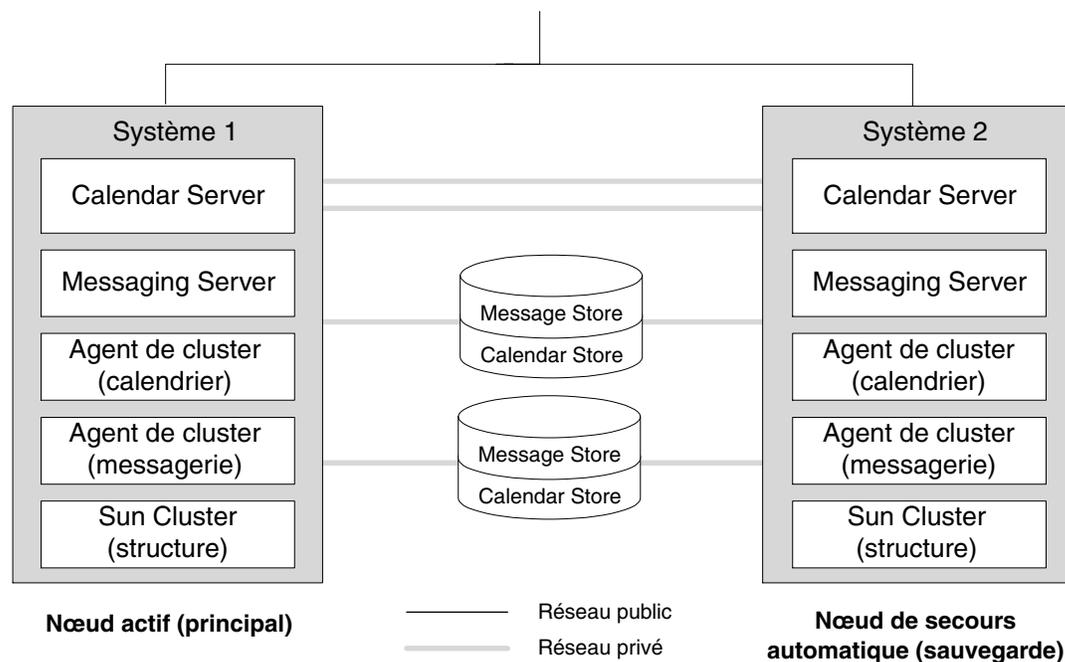


FIGURE 2-6 Conception de disponibilité à l'aide des nœuds Sun Cluster

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

Compte tenu du contrôle que procure le logiciel Sun Cluster, il 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 de répartir la demande accrue de ces services sur plusieurs instances simultanées. S'il est configuré convenablement, le logiciel Sun Cluster peut par conséquent assurer aussi bien une disponibilité élevée qu'une importante évolutivité 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 composants de Java ES, les services de disponibilité de Sun Cluster sont distribués sous la forme de services entre homologues. Le logiciel Sun Cluster doit donc être installé sur tous les ordinateurs d'un cluster.

Synthèse des trois dimensions architecturales

Considérées conjointement, les trois dimensions architecturales présentées dans la [Figure 2-1](#) et étudiées dans les sections précédentes forment une structure pour la conception de 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.

Toutefois, ce manuel n'a pas pour objectif de détailler les différentes méthodologies de conception basées sur la structure architecturale représentée par "[Structure architecturale de Java Enterprise System](#)" à la page 35. 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.

De nombreuses solutions peuvent être conçues et déployées en version standard, sans développement supplémentaire, en utilisant les composants fournis avec 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 fournit un exemple illustrant la prise en charge par Java Enterprise System d'une solution standard, bâtie autour des concepts d'architecture décrits dans 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 Récapitulatif des besoins d'entreprise : scénario de communications

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
Messagerie Calendrier	Messagerie électronique assurant la communication entre les employés et le monde extérieur. Dispositions électroniques des employés concernant le calendrier et les réunions.	Services de communication et de collaboration
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 suivante présente une architecture logique pour la fourniture de services de portail, de communication et d'identité identifiés dans le [Tableau 2-4](#) à l'aide de composants 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.

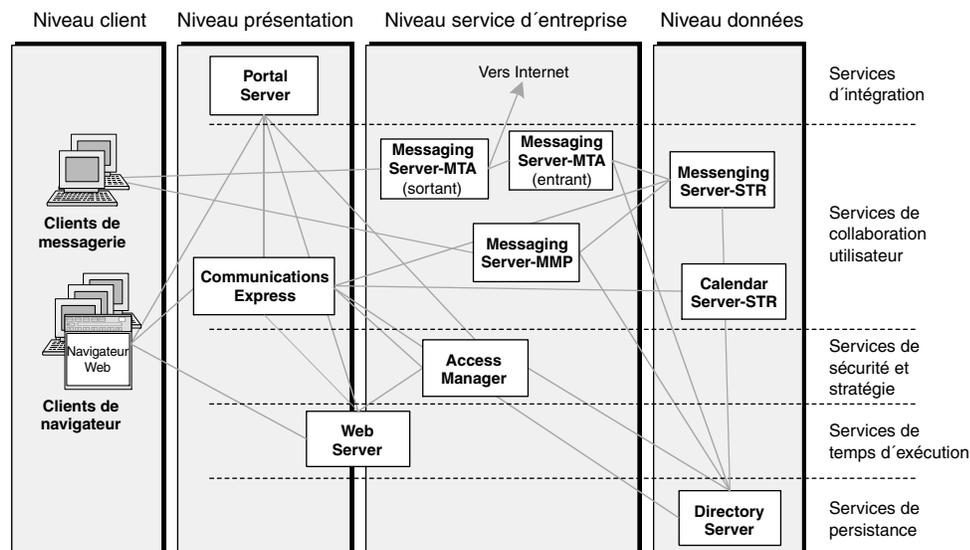


FIGURE 2-7 Architecture logique d'un scénario de communication d'entreprise

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 en tant que 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, en accédant aux informations utilisateur stockées dans Directory Server, régit l'autorisation et l'authentification de connexion unique pour Portal Server et les autres composants Web du niveau présentation. Les composants de Messaging Server incluent une mémoire de (Messaging Server-STR) au niveau données, l'envoi et la réception des composants au niveau services d'entreprise et 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. Portal Server, par exemple, dépend de Communications Express pour ses canaux de messagerie et de calendrier et 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 plus d'informations sur la conception logique d'une solution Java ES, reportez-vous au *Guide de planification du déploiement de Sun Java Enterprise System 2005Q4*.

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-feux 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 maître peut être utilisée pour Directory Server.

Pour plus d'informations sur la conception du déploiement d'une solution Java ES, reportez-vous au *Guide de planification du déploiement de Sun Java Enterprise System 2005Q4*.

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 exécutant une fonction de calcul particulière, fournissant ainsi des **services d'entreprise** aux **utilisateurs finals** ou à d'autres composants d'application. 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 encapsulés sous la forme de **services Web**.

architecture	Conception montrant les blocs constitutifs physiques et logiques d'une application distribuée (ou d'un autre système logiciel) et leurs relations les uns par rapport aux autres. Dans le cas d'une application d'entreprise distribuée , la conception architecturale inclut généralement à la fois l' architecture logique de l'application et l' architecture de déploiement
service d'entreprise	composant d'application ou assemblage de composants effectuant une logique d'entreprise pour le compte de clients multiples (et qui est donc par conséquent un processus multi-thread). Un service d'entreprise peut également être un assemblage de composants encapsulés sous la forme d'un service Web , ou encore être un serveur autonome.
client	Logiciel demandant des services logiciels. (Remarque : il ne s'agit pas d'une personne (voir utilisateur final .) Un client peut être un service envoyant une requête à un autre service, ou un composant de l'interface utilisateur graphique auquel accède un utilisateur final.
architecture de déploiement	Conception de haut niveau décrivant le mappage d'une architecture logique et d'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 décrivant les blocs constitutifs d'une application distribuée et les relations (ou interfaces) entre ces blocs. L'architecture logique comprend aussi bien les composants d'application distribués que les services d'infrastructure requis pour leur prise en charge.
serveur	Processus logiciel multi-thread (différent d'un serveur matériel) fournissant un service distribué ou un ensemble cohérent de services pour des clients accédant au service par le biais d'une interface externe.
service Web	Service conforme aux protocoles Internet standard en matière d'accessibilité, d'encapsulation de services et de détection. 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).

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 fonctionnalités suivantes :

- "Programme d'installation intégré de Java Enterprise System" à la page 55
- "Services intégrés d'identité et de sécurité" à la page 57
- "Termes clés de ce chapitre" à la page 61

Programme d'installation intégré de Java Enterprise System

Tous les composants de Java ES sont installés à l'aide d'un programme d'installation unique. 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 vers un système hôte. Il permet de sélectionner et d'installer un nombre quelconque de composants Java ES sur tout ordinateur de votre environnement informatique. Il assure également une configuration au moment de l'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 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 plus d'informations, reportez-vous au *Guide d'installation de Sun Java Enterprise System 2005Q4 pour UNIX*) :

- “Contrôle de logiciels préexistants” à la page 56
- “Contrôle de dépendance” à la page 56
- “Configuration initiale” à la page 57
- “Désinstallation” à la page 57

Contrôle de logiciels préexistants

Le programme d'installation examine l'ordinateur sur lequel l'installation est effectuée, puis il identifie les composants de 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 vérifie si des composants partagés de Java ES (voir “[Composants partagés](#)” à la page 26), comme J2SE ou NSS, sont déjà installés. 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 une vérification approfondie des composants afin de vérifier que les composants sélectionnés lors de l'installation fonctionneront bien conjointement.

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 entretenant des rapports de dépendance avec ce composant.

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 de 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 fournit 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 Java ES partagés.

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

L'une des fonctions importantes de Java Enterprise System est la gestion intégrée des identités des utilisateurs et la structure d'autorisation et d'authentification intégrée.

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” à la page 58
- “Authentification et connexion unique” à la page 59

Identité unique

Au sein d’un environnement Java ES, un utilisateur final possède une seule identité intégrée. En fonction de cette **identité unique**, un utilisateur peut être autorisé à accéder à certaines ressources, par exemple un portail, des pages Web, ou des services de messagerie, de calendrier ou de messagerie instantanée.

La fonctionnalité d’identité et de sécurité intégrée est basée sur la collaboration étroite entre Directory Server, Access Manager et les autres composants de Java ES.

L’accès d’un utilisateur à un service ou une ressource Java ES est conditionné par le stockage d’informations spécifiques à l’utilisateur dans une entrée utilisateur unique dans un référentiel utilisateur ou un **annuaire**. 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 annuaire fournis 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.

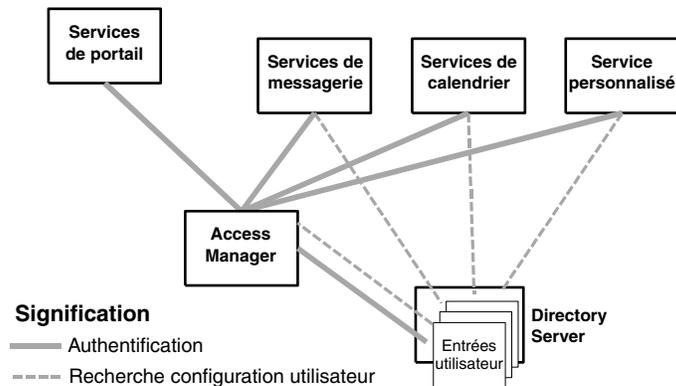


FIGURE 3-1 Une entrée utilisateur unique prend en charge de nombreux services

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

Authentification et connexion unique

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

Access Manager utilise également 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. L'agent de stratégie intercède auprès d'Access Manager pour les requêtes adressées par des utilisateurs aux ressources sécurisées. Pour certains composants Java ES, comme Portal Server et Communications Express, les fonctionnalités de l'agent de stratégie sont assurées par un sous-composant de Access Manager (voir "[Sun Java System Access Manager 7 2005Q4](#)" à la page 76).

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 peut pour cela utiliser un navigateur afin de consulter 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 dans 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.

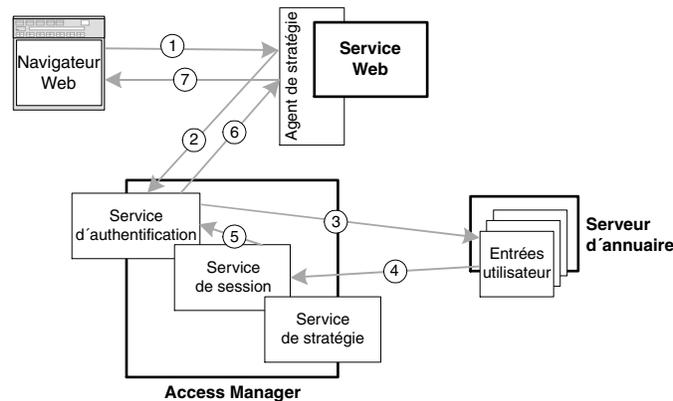


FIGURE 3-2 Séquence d'authentification

Connexion unique

Le scénario d'authentification traité dans les paragraphes précédents dissimule une étape importante. Lors de la vérification d'une demande d'authentification d'un utilisateur, le service de session de Access Manager est engagé (4), comme illustré dans 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 auprès du service de session que l'authentification précédente de l'utilisateur est toujours valide, puis l'utilisateur est autorisé à accéder au service pour la deuxième fois 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 inclut également un service de stratégie qui assure le contrôle d'accès aux ressources Web au sein d'un environnement Java ES. Une **stratégie** est une règle indiquant qui est autorisé à accéder à une ressource spécifique et dans quelles conditions. La séquence d'autorisation est illustrée à la figure suivante.

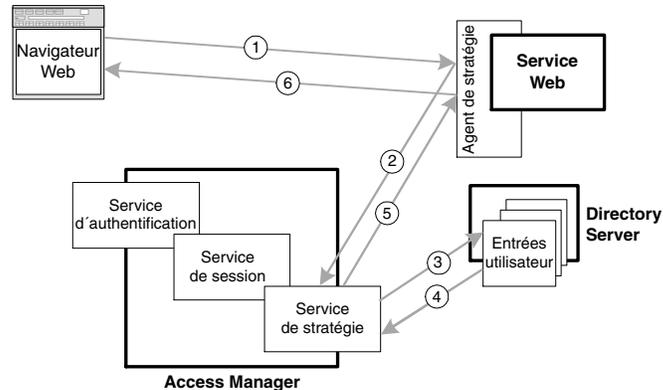


FIGURE 3-3 Séquence d'autorisation

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 fournit les moyens de définir, modifier, accorder, révoquer et supprimer des stratégies au sein d'une entreprise. Les stratégies sont stockées dans Directory Server et configurées par le biais d'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 Access Manager font appliquer les 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.

annuaire	Type particulier de base de données optimisé pour la lecture des données plutôt que pour leur écriture. La plupart des annuaires se basent sur le protocole standard LDAP (Lightweight Directory Access Protocol).
stratégie	Une stratégie est une règle indiquant qui est autorisé à accéder à une ressource donnée dans des conditions précises. Cette règle peut se baser sur des groupes d'utilisateurs ou des rôles au sein d'une organisation.
identité unique	Identité affectée à un utilisateur par le biais d'une entrée utilisateur unique dans un annuaire Java Enterprise System. Sur la base de cette entrée d'utilisateur unique, l'utilisateur peut obtenir l'accès à diverses ressources Java Enterprise System, telles que portails, pages Web et services (comme une messagerie, un calendrier et une messagerie instantanée, par exemple).
connexion unique	Fonction permettant l'authentification d'un utilisateur sur un service d'un système distribué et son application automatique aux autres services du système.

Tâches du cycle de vie de la solution Java Enterprise System

Ce chapitre décrit les concepts et la terminologie concernant chaque phase du cycle de vie de la solution Java ES. Il traite essentiellement des tâches de déploiement, notamment des tâches de conception et d'implémentation du déploiement.

Ce chapitre s'articule autour de trois groupes de phases du cycle de vie de la solution :

- "Déploiement préalable" à la page 65
- "Déploiement" à la page 66
- "Déploiement ultérieur" à la page 71
- "Termes clés de ce chapitre" à la page 72

Tâches du cycle de vie d'une solution

Le cycle de vie d'une solution a été présenté dans le [Chapitre 1](#), en tant qu'approche standard pour l'implémentation de solutions d'entreprise à l'aide du logiciel Java ES. Ce chapitre décrit les tâches intervenant à chaque phase du cycle de vie. Le diagramme du cycle de vie est repris dans la [Figure 4-1](#) pour faciliter les références.

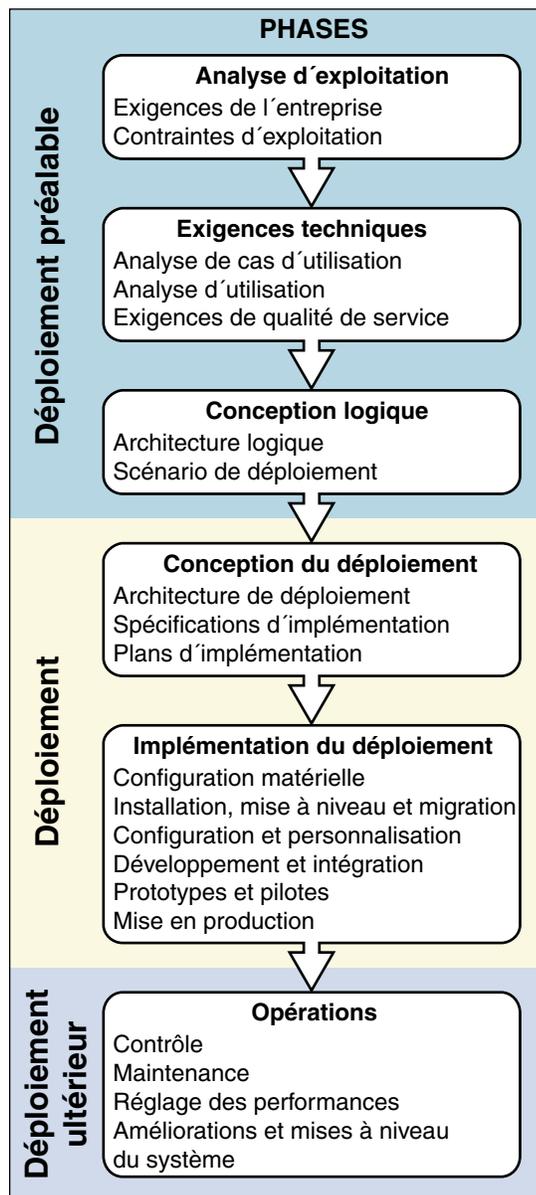


FIGURE 4-1 Tâches du cycle de vie d' une solution

Déploiement préalable

Au cours des phases de **déploiement préalable** du cycle de vie, vous transformez une analyse des besoins d'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 indiqué dans 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 des **cas d'utilisation** qui modélisent l'interaction de l'utilisateur avec un système logiciel anticipé. Vous déterminez également les modèles d'utilisation attendus pour ces cas d'utilisation. En conjuguant l'analyse d'exploitation et l'analyse d'utilisation, vous formulez les exigences de qualité de service (voir [Tableau 2-2](#)) auquel le déploiement proposé doit répondre.
- **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 de Java ES et ceux des applications personnalisées qui sont requis pour fournir les services d'utilisateur final. En utilisant les concepts expliqués dans le [Chapitre 2](#), 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, combinant performances, disponibilité, sécurité ainsi que d'autres exigences de qualité de service, est encapsulée dans un scénario de déploiement, comme indiqué dans la figure suivante. Pour plus d'informations sur les phases de déploiement préalable du cycle de vie, reportez-vous au *Guide de planification du déploiement de Sun Java Enterprise System 2005Q4*.

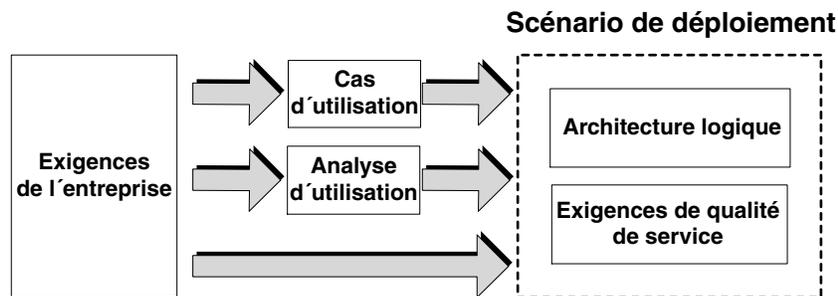


FIGURE 4-2 Spécification d'un scénario de déploiement

Déploiement

Au cours des phases de **déploiement** du cycle de vie, vous transformez un scénario de déploiement en un modèle de déploiement que vous allez ensuite implémenter, tester et mettre en production dans un environnement de 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 général, vous devez déployer à la fois les composants d'application personnalisés (composants J2EE, services Web ou autres serveurs) et les composants de Java ES requis pour la prise en charge de la solution.

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

- **“Conception du déploiement” à la page 67** : 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.
- **“Implémentation du déploiement” à la page 69** : l'implémentation 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

Au cours de la phase de conception du déploiement, vous créez une architecture de déploiement de niveau supérieur, suivie de spécifications de d'implémentation de niveau inférieur.

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.

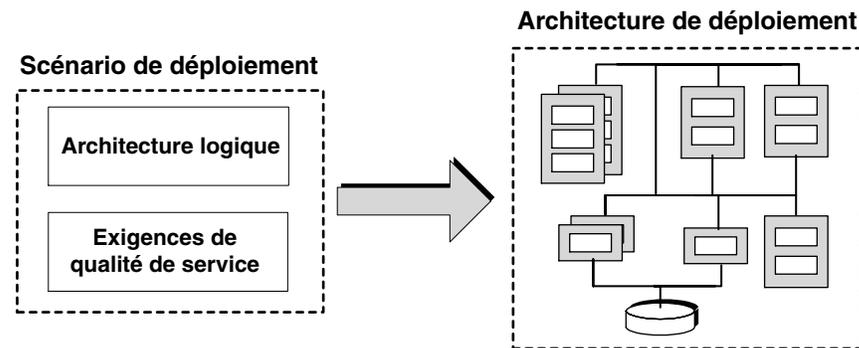


FIGURE 4-3 Conversion d'un scénario de déploiement en une architecture de déploiement

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) 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. Cependant, comme point de départ de la conception de déploiement, Java Enterprise System développe un ensemble d'[architecture de déploiement de référence](#).

Une architecture de référence est fondée 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 l'implémentation d'une architecture de déploiement. Parmi ces spécifications, on compte généralement les suivantes :

- matériel utilisé : ordinateurs, périphériques de stockage, équilibres de charge et 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 d'annuaire requises pour le provisioning des utilisateurs ;

Plans d'implémentation

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

- Configuration matérielle
- Installation, mise à niveau et migration logicielles

- Configuration et personnalisation du système
- Développement et intégration
- Test
- Production (démarrage)

Implémentation du déploiement

L'implémentation d'une conception de déploiement comprend les tâches répertoriées dans la section précédente et illustrées dans 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 l'implémentation du déploiement dans l'ordre selon lequel elles sont généralement effectuées. Pour une documentation de ces tâches, reportez-vous au *Présentation de la documentation de Sun Java Enterprise System 2005Q4*.

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 équilibreurs 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 utilisez le programme d'installation intégré de Java ES pour installer les composants Java ES appropriés sur chaque ordinateur de l'architecture de déploiement (voir "[Programme d'installation intégré de Java Enterprise System](#)" à la page 55).

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 provisionnés de sorte qu'ils puissent accéder à différents services, et des stratégies et contrôles d'authentification et d'autorisation doivent être mis en place (voir "[Services intégrés d'identité et de sécurité](#)" à la page 57).

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 allez généralement personnaliser Portal Server afin de fournir les canaux de portail, Access Manager pour réaliser les tâches d'autorisation et Messaging Server pour faire appel aux contrôles anti-virus et au filtrage anti-spam.

Développement et intégration

L'architecture logique spécifiée dans le scénario de déploiement détermine généralement l'étendue du travail de [développement](#) nécessaire pour implémenter 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 devez, à un moment donné, vérifier votre architecture de déploiement. En outre, il vous faut tester la solution par rapport aux cas d'utilisation afin de vous assurer 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 finals 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 important 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 finals 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 système : ces tâches incluent l'ajout de nouveaux composants Java ES à un système afin d'ajouter de nouvelles fonctionnalités ou de remplacer des composants non 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 Java ES possède ses propres outils d'administration permettant de configurer, d'ajuster ou de 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 cycle de vie d'une solution Java Enterprise System au cours de laquelle un scénario de déploiement est converti en conception de déploiement, implémenté, transformé en prototype et mis en production 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 pour une solution Java Enterprise System et exigences de qualité de service auxquelles la solution doit satisfaire pour répondre aux 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 d'une solution Java Enterprise System, au cours de laquelle les composants personnalisés d'une architecture de déploiement sont programmés et testés.
déploiement préalable	Phase du cycle de vie d'une solution Java Enterprise System au cours de laquelle les besoins d'exploitation sont convertis en un scénario de déploiement : une architecture logique et un ensemble d'exigences de qualité de service auxquelles une solution doit satisfaire.
déploiement ultérieur	Phase du cycle de vie d'une solution Java Enterprise System au cours de laquelle les applications distribuées sont mises en route, surveillées,

réglées de façon à optimiser les performances et mises à niveau de manière dynamique afin de prendre en compte les nouvelles fonctionnalités.

**architecture de
déploiement de
référence**

architecture de déploiement conçue, implémentée et dont les performances ont été testées. 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 ou ensemble de tâches utilisateur spécifiques effectuées par une **application d'entreprise distribuée**, et utilisées comme point de départ pour la conception, le test et la mesure des performances de l'application.

Liste de référence : composants Java Enterprise System

Cette annexe présente une liste de références de tous les composants de Java ES, regroupés dans les catégories suivantes :

- [“Description des composants de service du système” à la page 76](#). 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 [“À quoi sert Java Enterprise System ?” à la page 19](#), comprennent les services de portail, les services de communication et de collaboration, les services d’identité et de sécurité, les services Web et d’application et les services de disponibilité.
- [“Description des composants de qualité de service” à la page 80](#). 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” à la page 85](#). Ces composants sont des bibliothèques locales pouvant être partagées par tout composant de Java ES fonctionnant sur un ordinateur hôte donné.

Dans cette annexe, les composants de Java ES sont présentés par ordre alphabétique au sein de leurs catégories et sous-catégories respectives.

Pour une présentation de la documentation afférente à chaque composant, reportez-vous à *Présentation de la documentation de Sun Java Enterprise System 2005Q4*.

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 7 2005Q4" à la page 76
- "Sun Java System Application Server Enterprise Edition 8.1 2005Q4" à la page 77
- "Sun Java System Calendar Server 6 2005Q4" à la page 77
- "Sun Java System Directory Server 5 2005Q4" à la page 78
- "Sun Java System Instant Messaging 7 2005Q4" à la page 78
- "Sun Java System Message Queue 3.6 2005Q4" à la page 78
- "Sun Java System Messaging Server 6 2005Q4" à la page 79
- "Sun Java System Portal Server 6 2005Q4" à la page 79
- "Sun Java System Service Registry 3 2005Q4" à la page 80
- "Sun Java System Web Server 6.1 2005Q4" à la page 80

Sun Java System Access Manager 7 2005Q4

Sun Java System Access Manager (Access Manager) fournit une infrastructure permettant à une organisation de gérer les processus utilisés pour administrer les identités numériques des clients, employés et partenaires utilisant leurs services Web et leurs applications non 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 de 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 inclut également les sous-composants SDK Access Manager et Delegated Administrator (voir "Sun Java System Delegated Administrator 6 2005Q4" à la page 84).
- **Access Manager SDK** : offre une interface distante vers Access Manager. Ce sous-composant doit être installé sur tout ordinateur hébergeant un composant Java ES accédant à distance à Access Manager.
- **Access Manager Administration Console** : cette interface graphique regroupe les services d'identité et la gestion de stratégies et offre aux utilisateurs une interface unique pour créer et gérer leur compte, les attributs de service ainsi que les règles d'accès dans 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 8.1 2005Q4

Sun Java System Application Server (Application Server) fournit une plate-forme compatible J2EE pour le développement et le déploiement des services d'application et des services Web. Application Server apporte les services d'infrastructure permettant l'interaction entre des composants distribués à configuration groupée, y compris l'appel de méthode distant et d'autres services 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 de 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 2005Q4

Sun Java System Calendar Server (Calendar Server) est une solution évolutive basée sur le Web, utilisée pour la gestion centralisée des calendriers et la planification à l'intention des entreprises et fournisseurs de services. Calendar Server prend en charge les calendriers personnels et de groupe, mais aussi de ressources, telles que les salles de conférence ou le matériel.

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

Sun Java System Directory Server 5 2005Q4

Sun Java System Directory Server (Directory Server) offre un service d'annuaire centralisé pour les informations relatives à vos réseaux (intranet et/ou extranet). Directory Server s'intègre aux systèmes existants et fait office de référentiel centralisé par le regroupement des informations sur vos employés, clients, fournisseurs et partenaires. Vous pouvez étendre Directory Server de manière à gérer les préférences et profils des utilisateurs, ou encore 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 2005Q4

Sun Java System Instant Messaging (Instant Messaging) permet aux utilisateurs de participer à des sessions de messagerie instantanée et de discussion, de s'envoyer réciproquement des messages d'alerte, et de partager instantanément des forums groupés. Instant Messaging convient à la fois aux intranets et à Internet, et autorise les interactions avec d'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 3.6 2005Q4

Sun Java System Message Queue (Message Queue) est une solution basée sur des normes permettant de résoudre la problème des communications inter-applications et d'assurer une livraison fiable des messages. Message Queue est un système de messagerie d'entreprise appliquant la norme ouverte Java Message Service (JMS).

En plus d'être un fournisseur JMS, Message Queue possède des fonctionnalités dépassant les exigences minimales de la spécification JMS. Grâce au logiciel Message Queue, les processus exécutés sur différentes plates-formes et différents systèmes d'exploitation peuvent se connecter à un service de messagerie Message Queue commun pour l'envoi et la réception d'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.

Deux éditions de Message Queue sont disponibles :

- **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.

Les composants Message Queue Enterprise Edition et Message Queue Platform Edition fournis par le programme d'installation de Java ES peuvent être installés séparément.

Sun Java System Messaging Server 6 2005Q4

Sun Java System Messaging Server (Messaging Server) est un serveur de messagerie Internet puissant, basé sur des normes, destiné aux entreprises et fournisseurs de services. Conçu pour assurer une gestion fiable et de haute capacité des messages, Messaging Server se compose de plusieurs composants modulaires, configurables indépendamment les uns des autres, offrant une prise en charge de plusieurs protocoles de messagerie électronique.

Le programme d'installation de Java ES fournit Messaging Server sous la forme d'un composant unique installable. Toutefois, après l'installation, chaque instance peut être configurée en vue de fournir n'importe lequel des nombreux services de messagerie, représentant 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 2005Q4

Sun Java System Portal Server (Portal Server) est une solution de serveur de portail compatible avec les identités. Portal Server combine des services de portail essentiels, 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, avec Mobile access, sous la forme d'un composant unique installable.

Sun Java System Service Registry 3 2005Q4

Sun Java System Service Registry (Service Registry) est un référentiel jouant à la fois le rôle de registre de services Web (UDDI) et de registre XML d'entreprise (ebXML) pour la prise en charge des applications dont l'architecture est orientée vers les services Web (SOA). Le registre UDDI est utilisé pour enregistrer et détecter les services Web, tandis que le registre ebXML permet de stocker et de gérer les artefacts d'informations nécessaires pour prendre en charge certains processus d'entreprise. Ces artefacts comprennent les métadonnées telles que le schéma XML, les règles des processus d'entreprise, les contrôles d'accès aux services Web, les contrôles de version, les schémas de classification, etc.

Sun Java System Web Server 6.1 2005Q4

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 offre performances, fiabilité, évolutivité et d'excellentes capacités de gestion pour toutes les entreprises, quelle que soit leur taille. Web Server prend en charge un grand nombre de standards logiciels Web, parmi lesquels JDK 1.4.1, Java Servlet 2.3, JavaServer Pages™ (JSP™) 1.2, HTTP/1.1, PKCS #11, FIPS-140, les certificats 168 bits, ainsi que de nombreux autres standards 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.

Ces composants sont décrits dans les sections suivant celle-ci.

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 8/05 et Sun Cluster Agents”](#) à la page 81
- [“High Availability Session Store 2005Q4”](#) à la page 81

Sun Cluster 3.1 8/05 et Sun Cluster Agents

Le logiciel Sun Cluster fournit des services haute disponibilité Évolutivité pour Java Enterprise System, et pour les applications basées sur l'infrastructure de Java ES.

Un cluster est un ensemble d'ordinateurs interconnectés (nœuds du cluster) fournissant conjointement une vue client unique des services, ressources du système et 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 surveille en permanence le bon fonctionnement des nœuds membres et autres ressources du cluster et utilise la redondance interne pour assurer un accès quasi-continu à ces ressources, même en cas d'erreur.

Le programme d'installation de Java ES fournit le noyau Sun Cluster et les agents Sun Cluster sous la forme de composants installables séparément. Des agents Sun Cluster supplémentaires sont disponibles sur des CD distincts.

High Availability Session Store 2005Q4

Sun Java System High Availability Session Store (HADB) fournit un magasin de données qui peut être utilisé pour préserver la disponibilité des données 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, souvent à partir de 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 6 2005Q4" à la page 82
- "Sun Java System Web Proxy Server 4.0.1 2005Q4" à la page 82
- "Sun Java System Connector pour Microsoft Outlook 6 2005Q4" à la page 82
- "Sun Java System Directory Proxy Server 5 2005Q4" à la page 83
- "Sun Java System Portal Server Secure Remote Access 6 2005Q4" à la page 83

Sun Java System Communications Express 6 2005Q4

Sun Java System Communications Express (Communications Express) fournit un client de communications et de collaboration intégré basé sur le Web offrant aux utilisateurs une interface Web avec services de calendrier, de messagerie et de carnet d'adresses. Communications Express se compose des trois modules client suivants : calendrier, carnet d'adresses et courrier. Configurable de manière à donner accès à Messaging Server, Calendar Server ou les deux, Communications Express fonctionne avec Sun Java System LDAP Schéma, 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 Web Proxy Server 4.0.1 2005Q4

Sun Java System Web Proxy Server (Web Proxy Server) assure la mise en cache, le filtrage et la distribution de contenus Web. Web Proxy Server est souvent utilisé au sein des pare-feux d'entreprise afin de réduire le nombre de requêtes vers les serveurs de contenu distants et à l'extérieur des pare-feux afin de fournir une passerelle de sécurité pour les requêtes Internet entrantes.

Le programme d'installation de Java ES fournit Web Proxy Server en tant que composant installable unique.

Sun Java System Connector pour Microsoft Outlook 6 2005Q4

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

Connector for Microsoft Outlook envoie des requêtes à Messaging Server pour s'enquérir des hiérarchies de dossiers et des messages électroniques, puis convertit ces informations en propriétés Messaging API (MAPI) pouvant être affichées par Outlook. De même, il utilise le protocole WCAP pour interroger Calendar Server pour les

événements et tâches qui sont ensuite convertis en propriétés MAPI. Grâce à ce modèle, Sun Java System Connector pour Microsoft Outlook construit un affichage d'utilisateur final Outlook à partir de deux sources d'informations séparées: la messagerie à partir de Messaging Server et les informations d'agenda à partir 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 2005Q4

Sun Java System Directory Proxy Server (Directory Proxy Server) est un composant essentiel de tout service d'annuaire pour les solutions de commerce électronique. Directory Proxy Server est une passerelle LDAP de la couche application qui offre un contrôle des accès aux annuaires amélioré, une compatibilité de schémas et une haute disponibilité grâce aux fonctions de basculement et d'équilibrage de charge.

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 2005Q4

Sun Java System Portal Server Secure Remote Access (Portal Server Secure Remote Access) étend Portal Server en proposant un accès sécurisé distant via un navigateur au contenu et aux services de Portal Server à partir de n'importe quel navigateur distant, éliminant ainsi la nécessité de recourir au logiciel client. Son intégration à Portal Server garantit aux utilisateurs l'utilisation d'un accès sécurisé au contenu et aux services auxquels ils ont 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.
- **Proxy netlet :** 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.
- **Serveur proxy de réécriture :** 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 la console correspondante) 5 2005Q4” à la page 84
- “Sun Java System Directory Preparation Tool 2005Q4” à la page 84
- “Sun Java System Delegated Administrator 6 2005Q4” à la page 84

Sun Java System Administration Server (et la console correspondante) 5 2005Q4

Sun Java System Administration Server (Administration Server) et Server Console fournissent conjointement un outil graphique permettant de gérer Directory Server, Directory Proxy Server et Messaging Server. Administration Server traite les requêtes des serveurs d'un groupe de serveurs placés dans un même répertoire racine, puis lance les programmes indispensables à la réalisation de ces requêtes.

Server Console est une application Java autonome qui fonctionne en combinaison avec une instance de Directory Server et une instance d'Administration Server sur votre réseau. Server Console fait office d'application de gestion frontale des logiciels Java ES de votre entreprise.

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

Sun Java System Directory Preparation Tool 2005Q4

Sun Java System Directory Preparation Script est utilisé pour configurer Directory Server à l'aide du schéma requis pour provisionner les utilisateurs pour Messaging Server et Calendar Server.

Sun Java System Delegated Administrator 6 2005Q4

Sun Java System Delegated Administrator est un utilitaire de ligne de commande (`comadmin`) servant au provisioning des utilisateurs, groupes, domaines et ressources pour Calendar Server, Messaging Server, ainsi que d'autres fournisseurs de services Java ES.

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

Composants partagés

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

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

- ANT (outil intégré basé sur Jakarta ANT Java/XML)
- Apache Commons Logging
- Berkeley DB
- Conteneur d'agent commun
- ICU (International Components for Unicode)
- J2SE™ (Java 2 Platform, Standard Edition)
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Enterprise Web Application Framework)
- JavaHelp™ Runtime
- JavaMail™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries) Runtime
- JAX-RPC (Java API for XML-based Remote Procedure Call) Runtime
- JCAPI (Java Calendar API)
- JDMK (Java Dynamic Management™ Kit) Runtime
- JSS (Java Security Services)
- KTSE (KT Search Engine)
- LDAP C SDK
- LDAP Java SDK
- MA (Mobile Access) Core
- MFWK (Java ES Monitoring Framework)
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- SAAJ (SOAP with Attachments API for Java)
- SASL (Simple Authentication and Security Layer)
- Sun Explorer Data Collector (Solaris uniquement)
- Sun Java Web Console
- WSCL (Web services Common Library)

Index

A

Access Manager
 Composant de service système, 23
 Description, 76
 Service d'infrastructure, 40
Administration Server
 Composant de qualité de service, 26
 Description, 84
Améliorations, *Voir* Scénarios d'adoption
Annuaire
 Définition, 62
 sous la forme de magasins de données
 utilisateur, 58
Ant (outil intégré basé sur Jakarta ANT
 Java/XML), 85
Application Server
 Composant de service système, 23
 Description, 77
 Service d'infrastructure, 40
Applications
 Distribuées
 Voir Applications d'entreprise distribuées
 Entreprise
 Voir Applications d'entreprise distribuées
Applications d'entreprise distribuées
 À propos de, 19
 Définition, 33
 Infrastructure, 20
Architecture
 Définition, 54
 Déploiement, 67-68
 Dimensions
 Voir Dimensions architecturales

Architecture (Suite)
 Présentation, 35
 Référence, 68
 Solution, 36
Architectures de déploiement
 Conception, 67-68
 Définition, 54
 présentation, 35
 Relation avec les architectures à plusieurs
 niveaux, 44
Architectures de déploiement de référence
 Définition, 73
 Présentation, 68
Architectures logiques
 Définition, 54
 Exemple, 52-53
 Niveaux de service de l'infrastructure, 37
 Présentation, 35
Authentification, 59-60
Autorisation, 61

B

Berkeley DB, 85

C

Calendar Server
 Composant de service système, 23
 Description, 77
 Service d'infrastructure, 40

- CAPI (Java Calendar API), 85
- Cas d'utilisation
 - Définition, 73
 - Présentation, 65
- Catégories d'utilisateur
 - Administrateur délégué, 30
 - Administrateur système, 30
 - Administrateur système spécialisé, 30
 - Analyste système, 29
 - Architecte, 29
 - Ingénieur de projet, 30
 - Intégrateur système, 30
 - Planificateur, 29
 - Responsable informatique, 30
- Client, niveau logique, 43
- Clients
 - Composants de services système, 22
 - Définition, 54
- Clusters
 - Voir* Sun Cluster
- Communications Express
 - Composant de qualité de service, 25
 - Description, 82
- Components, Qualité de service, 80-84
- Composants
 - Accès, 25-26
 - Administratifs, 26
 - Définition, 32
 - Dépendances, 41-42
 - Détection des versions installées, 56
 - Disponibilité, 24-25
 - Distribués, 19
 - EJB, 43
 - J2EE, 43
 - JSP, 43
 - Partagés, 26, 85
 - Qualité de service, 24-26
 - Service du système, 76-80
 - Service système, 22-24
 - Services d'infrastructure, 40
 - Servlet, 43
 - Système
 - Voir* Composants système
- Composants administratifs
 - Description, 84
 - Présentation, 26
- Composants d'accès
 - Description, 82-83
- Composants d'accès (Suite)
 - Présentation, 25-26
- Composants d'application
 - Dans les architectures de niveaux logiques, 43
 - Définition, 53
- Composants de disponibilité
 - Description, 81
 - Présentation, 24-25
- Composants de qualité de service
 - Définition, 33
 - Descriptions, 80-84
 - Présentation, 24-26
- Composants de service du système,
 - Définition, 33
- Composants de service système
 - Dépendances, 41-42
 - Présentation, 22-24
- Composants EJB, 43
- Composants Java Servlet, 43
- Composants JSP, 43
- Composants partagés, 85
 - Définition, 33
 - Présentation, 26
- Composants système
 - À propos de, 22
 - Composants de qualité de service, 24-26
 - Composants de service système
 - Voir* Composants de service du système
 - Composants partagés, 26, 85
 - Définition, 33
 - Service du système, 76-80
- Connector for Microsoft Outlook
 - Composant de qualité de service, 25
 - Description, 82
- Connexion unique
 - Définition, 62
 - Fonction Java ES, 23, 59
 - Implémentation, 60
 - Niveaux de service d'infrastructure, 39
- Conteneur d'agent commun, 85
- Contrôle de dépendance, programme d'installation, 56-57
- Cycle de vie, phases
 - Déploiement, 29, 66
 - Déploiement préalable, 29, 65
 - Déploiement ultérieur, 29

D

- Delegated Administrator
 - Composant de qualité de service, 26
 - Description, 84
- Dépendances, 41-42, 57
- Déploiement
 - Architecture, 67
 - Architectures de référence, 68
 - Conception, 67-69
 - Définition, 72
 - Développement et personnalisation, 70
 - Implémentation, 69-71
 - Mise en production, 71
 - Phases du cycle de vie, 66-71
 - Scénarios
 - Voir* Scénarios de déploiement
 - Test de prototype, 70
- Déploiement préalable
 - Définition, 72
 - Phases du cycle de vie, 65
- Déploiement ultérieur
 - Définition, 72
 - Phases du cycle de vie, 71
- Détection des logiciels installés, 56
- Développement
 - Définition, 72
 - Tâche de déploiement, 70
- Dimensions architecturales
 - Dépendances des services d'infrastructure, 37
 - Niveaux logiques, 42
 - Qualité de service, 45-49
 - Synthèse, 50
- Directory Preparation Tool
 - Composant de qualité de service, 26
 - Description, 84
- Directory Proxy Server
 - Composant de qualité de service, 25
 - Composant système, 47
 - Description, 83
- Directory Server
 - Composant de service système, 23
 - Description, 78
 - Service d'infrastructure, 40
- Disponibilité
 - Exigences, 46, 47, 48
 - Services, 48, 81

Distribuées

- Applications
 - Voir* Applications d'entreprise distribuées
- Distribués
 - Services
 - Voir* Services distribués

E

- Entrée utilisateur, 58
- Évolutivité
 - Exigences, 46, 47, 48
 - Services, 48, 81
- Exigences de capacité latente, 46
- Exigences de facilité de maintenance, 46, 48
- Exigences de performance, 46
- Exigences de qualité de service
 - Capacité latente, 46
 - Disponibilité, 46, 47, 48
 - Évolutivité, 46, 47, 48
 - Facilité de maintenance, 46
 - Performances, 46
 - Sécurité, 46, 47
- Exigences e qualité de service, Facilité de maintenance, 48

F

- Fonctions d'intégration
 - Composants partagés, 21
 - Identité et sécurité, 21, 57-61
 - Présentation, 21
 - Programme d'installation intégré, 21, 55-57
- Formation, scénarios d'adoption de Java ES, 31

G

- Glossaire, lien, 15

H

- High Availability Session Store
 - Composant de qualité de service, 25
 - Description, 81

I

- ICU (International Components for Unicode), 85
- Identité
 - Gestion, 57
 - Services, 21, 57-61
 - Utilisateur unique, 58-59
- Identité unique
 - Définition, 62
 - Présentation, 58
- Infrastructure
 - Applications d'entreprise distribuées, 20
 - Dépendances des services
 - Voir Services distribués*
- Instant Messaging
 - Composant de service système, 23
 - Description, 78
 - Service d'infrastructure, 40
- Intégration
 - Fonctions
 - Voir Fonctions d'intégration*
 - Scénarios d'adoption de Java ES, 31
 - Services, 39

J

- J2EE
 - Composants, 43
 - Modèle de composant distribué, 43
 - Plate-forme, 24
- J2SE (Java 2 Platform, Standard Edition), 26, 85
- JAF (JavaBeans™ Activation Framework), 85
- JATO (Java Studio Enterprise Web Application Framework), 85
- JavaHelp, 85
- JavaMail, 85
- JAX-RPC, 85
- JAXB (Java Architecture for XML Binding), 85
- JAXP (Java API for XML Processing), 85
- JAXR (Java API for XML Registries), 85
- JDMK (Java Dynamic Management™ Kit), 85
- JMS (Java Message Service), 23
- JSS (Java Security Services), 26, 85

K

- KT Search Engine (KTSE), 85

L

- Langues prises en charge, 56
- LDAP, 44, 62
- LDAP C SDK, 85
- LDAP Java SDK, 85
- Linux, 40

M

- Matériel, scénarios d'adoption de Java ES, 31
- Message Queue
 - Composant de service système, 23
 - Description, 78
 - Service d'infrastructure, 40
- Messaging Server
 - Composant de service système, 24
 - Description, 79
 - Service d'infrastructure, 40
- Migration, scénarios d'adoption de Java ES, 31
- Mise en production, 71

N

- Niveau logique
 - Client, 43
 - Données, 43
 - Présentation, 43
 - Service d'entreprise, 43
- Niveaux, logiques, Architecture d'application, 42
- NSPR (Netscape Portable Runtime), 26, 85
- NSS (Network Security Services), 26, 85

P

- Phases du cycle de vie, Déploiement ultérieur, 71
- Plate-forme J2ME, 43
- Portal Server
 - Composant de service système, 24
 - Description, 79
 - Service d'infrastructure, 40
- Portal Server Secure Remote Access
 - Composant de qualité de service, 25
 - Composant système, 47

Portal Server Secure Remote Access (Suite)
 Description, 83
 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 for Microsoft Outlook
 Voir Connector for 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
 Voir Portal Server
 Portal Server, Secure Remote Access
 Voir Portal Server, Secure Remote Access
 Web Server
 Voir Web Server
 Profils utilisateur, 29
 Programme de désinstallation, 57
 Prototype, 70
 Provisioning d'utilisateurs, 68

S

SAAJ (SOAP with Attachments API for
 Java), 85

SASL (Simple Authentication and Security
 Layer), 85
 Scénarios d'adoption, Java ES
 À propos de, 30-32
 Amélioration, 30
 Définition, 32
 Extension, 30
 Mise à niveau, 31
 Nouveau système, 30
 Scénarios de déploiement
 Définition, 72
 Présentation, 65
 Sécurité
 Exigences, 46, 47
 Services, 21
 Services de stratégie, 39
 Serveurs
 Autonomes, 43
 Définition, 54
 Service, Web, 43
 Services
 Définition, 33
 Évolutivité, 48, 81
 Haute disponibilité, 48, 81
 Infrastructure, 20
 Voir Services d'infrastructure distribués
 Services administratifs, 21
 Services d'accès, 21
 Services d'application, 21, 37
 Services d'entreprise
 Définition, 54
 Niveau présentation, 43
 Services d'exécution, 39
 Services de collaboration, 20
 Services de collaboration utilisateur, 39
 Services de communication, 20
 Services de disponibilité, 21
 Services de messagerie, 38
 Services de persistance, 38
 Services de plate-forme, 37
 Services de portail, 20
 Services de transport réseau, 38
 Services distribués
 Accès, 21
 Administratifs, 21
 Collaboration utilisateur, 39
 Communication et collaboration, 20
 Disponibilité, 21

- Services distribués (Suite)
 - Exécution, 21, 39
 - Identité, 21
 - Infrastructure, 20
 - Intégration, 39
 - Intermédiaires, 37
 - Messagerie, 38
 - Niveau application, 37
 - Persistence, 38
 - Plate-forme, 37, 38
 - Portail, 20
 - Présentation, 20
 - Sécurité, 21, 39
 - Transport réseau, 38
 - Web, 21
- Services du système d'exploitation, 38
- Services intermédiaires, 37
- Services système
 - À propos de, 22
 - Définition, 33
- Services Web, 21
 - Composants J2EE, 43
 - Définition, 54
- SOAP (Apache), 85
- SOAP Apache, temps d'exécution, 85
- Solaris, 40
- Solutions, Java ES
 - Architecture, 35
 - Cycle de vie, 27-30
 - Exemple, 51
 - Personnalisées et standard, 50
- Spécifications d'implémentation, 68
- Stratégies, Autorisation, 61
- Stratégies, Définition, 62
- Sun Cluster
 - Agents, 49
 - Composant de qualité de service, 25
 - Description, 81
 - Service de disponibilité, 48-49
- Sun Explorer Data Collector, 85
- Sun Java Web Console, 85
- Système
 - Composants
 - Voir Composants système*
 - Configuration, 57
 - Services, 19-21

T

- Tâches, Java Enterprise System, 27, 63

U

- Utilisateurs, Applications distribuées, 20
- Utilisateurs finals, Définition, 33

W

- Web Server
 - Composant de service système, 24
 - Description, 80
 - Service d'infrastructure, 40
- WSCL (Web services Common Library), 85