



# Présentation technique de Sun Java Enterprise System 5 Update 1



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

Référence : 820-3092-10  
Septembre 2007

Sun Microsystems, Inc. détient les droits de propriété intellectuelle de la technologie utilisée par le produit décrit dans le présent document. En particulier, et sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou dépôts de brevets en cours d'homologation aux États-Unis et dans d'autres pays.

Droits du gouvernement américain – Logiciel commercial. Les utilisateurs gouvernementaux sont soumis au contrat de licence standard de Sun Microsystems, Inc. et aux clauses applicables du FAR et de ses suppléments.

Cette distribution peut intégrer des éléments conçus par des tiers.

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, exclusivement concédée en licence par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Toutes les marques déposées SPARC sont utilisées sous licence et sont des marques commerciales ou déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont constitués selon une architecture développée par Sun Microsystems, Inc.

L'interface utilisateur graphique OPEN LOOK et Sun<sup>TM</sup> 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 sur l'interface d'utilisateur graphique Xerox, cette licence couvrant également les détenteurs de licence Sun qui mettent en place l'interface utilisateur graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits cités dans la présente publication et les informations qu'elle contient sont soumis à la législation américaine relative au contrôle sur les exportations et, le cas échéant, aux lois sur les importations ou exportations dans d'autres pays. Il est strictement interdit d'employer ce produit conjointement à des missiles ou armes biologiques, chimiques, nucléaires ou de marine nucléaire, directement ou indirectement. Il est strictement interdit d'effectuer des exportations et réexportations vers des pays soumis à l'embargo américain ou vers des entités identifiées sur les listes noires des exportations américaines, notamment les individus non autorisés et les listes nationales désignées.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, REPRÉSENTATIONS ET GARANTIES EXPRESSES OU TACITES, Y COMPRIS TOUTE GARANTIE IMPLICITE RELATIVE À LA COMMERCIALISATION, L'ADÉQUATION À UN USAGE PARTICULIER OU LA NON-VIOLATION DE DROIT, SONT FORMELLEMENT EXCLUES. CETTE EXCLUSION DE GARANTIE NE S'APPLIQUERAIT PAS DANS LA MESURE OÙ ELLE SERAIT TENUE JURIDIQUEMENT NULLE ET NON AVENUE.

# Table des matières

---

<b>Préface</b> .....	11
<b>1 Présentation de Java Enterprise System</b> .....	17
À quoi sert Java Enterprise System ? .....	17
Composants Java ES .....	19
Composants de services système .....	20
Composants de qualité de service .....	22
Composants partagés .....	23
Composants des Suites Sun Java .....	24
Utilisation de Java ES .....	26
Cycle de vie d'une solution Java ES .....	26
Scénarios d'adoption de Java ES .....	29
Termes clés de ce chapitre .....	31
<b>2 Architectures de solutions Java ES</b> .....	33
Structure architecturale Java ES .....	33
Dimension 1 : dépendances des services d'infrastructure .....	34
Dimension 2 : niveaux logiques .....	40
Dimension 3 : qualité de service .....	43
Synthèse des trois dimensions architecturales .....	47
Exemple d'architecture de solution Java ES .....	47
Scénario de communication d'entreprise .....	48
Architecture logique de l'exemple de scénario .....	48
Architecture de déploiement de l'exemple de scénario .....	50
Termes clés de ce chapitre .....	50

---

<b>3 Fonctions d'intégration de Java ES</b> .....	53
Programme d'installation intégré de Java ES .....	53
Contrôle de logiciels préexistants .....	54
Contrôle de dépendance .....	54
Configuration initiale .....	54
Désinstallation .....	55
Services de contrôle système .....	55
Services intégrés d'identité et de sécurité .....	55
Identité unique .....	56
Authentification et connexion unique .....	57
Autorisation .....	58
Termes clés de ce chapitre .....	59
<b>4 Cycle de vie d'une solution Java ES</b> .....	61
Tâches du cycle de vie d'une solution .....	61
Prédéploiement .....	63
Déploiement .....	64
Conception de déploiement .....	64
Implémentation du déploiement .....	66
Postdéploiement .....	69
Termes clés de ce chapitre .....	69
<b>A Composants Java ES</b> .....	71
Composants de services système .....	71
Access Manager 7.1 .....	72
Application Server Enterprise Edition 8.2 .....	73
Directory Server Enterprise Edition 6.2 .....	73
Java DB 10.2 .....	74
Message Queue 3.7 UR2 .....	74
Portal Server 7.1 Update 2 .....	74
Service Registry 3.1 .....	75
Web Server 7.0 .....	75
Composants de qualité de service .....	75
Composants de disponibilité .....	76
Composants d'accès .....	78

---

Composants de contrôle .....	79
Composants partagés .....	79
<b>Index</b> .....	<b>81</b>



# Liste des tableaux

---

TABLEAU 1-1	Composants de service système Java ES .....	21
TABLEAU 1-2	Composants de disponibilité de Java ES .....	22
TABLEAU 1-3	Composants d'accès de Java ES .....	23
TABLEAU 1-4	Composants des Suites Sun Java .....	24
TABLEAU 1-5	Catégories d'utilisateur de Java ES pour les tâches du cycle de vie .....	28
TABLEAU 1-6	Problèmes relatifs aux scénarios d'adoption de Java ES .....	30
TABLEAU 2-1	Relations entre les composants de service système Java ES .....	39
TABLEAU 2-2	Qualités de service affectant l'architecture de la solution .....	44
TABLEAU 2-3	Composants de qualité de service et qualités système influencées .....	45
TABLEAU 2-4	Récapitulatif des besoins d'entreprise : scénario de communications .....	48



# Liste des figures

---

FIGURE 1-1	Prise en charge requise pour les applications d'entreprise distribuées .....	18
FIGURE 1-2	Catégories de composants Java ES .....	20
FIGURE 1-3	Phases du cycle de vie d'une solution et catégories d'utilisateurs .....	27
FIGURE 2-1	Dimensions de l'architecture d'une solution Java ES .....	34
FIGURE 2-2	Dimension 1 : niveaux des services d'infrastructure .....	36
FIGURE 2-3	Composants de service système Java ES .....	38
FIGURE 2-4	Dimension 2 : niveaux logiques pour les applications d'entreprise distribuées .....	40
FIGURE 2-5	Messaging Server : exemple d'architecture à plusieurs niveaux .....	43
FIGURE 2-6	Architecture logique d'un scénario de communication d'entreprise .....	49
FIGURE 3-1	Une entrée utilisateur unique prend en charge de nombreux services .....	56
FIGURE 3-2	Séquence d'authentification .....	58
FIGURE 3-3	Séquence d'autorisation .....	59
FIGURE 4-1	Tâches du cycle de vie d'une solution .....	62
FIGURE 4-2	Spécification d'un scénario de déploiement .....	63
FIGURE 4-3	Conversion d'un scénario de déploiement en une architecture de déploiement .....	65



# Préface

---

La *Présentation technique de Sun Java Enterprise System 5 Mise à jour*aborde les concepts et informations techniques de Sun Java™ Enterprise System (Java ES). Elle présente également les composants de Java ES, 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 dernière section de chaque chapitre.

## À qui ce manuel s'adresse-t-il ?

Ce manuel s'adresse aux personnes chargées de la conception, du déploiement ou de la maintenance de solutions logicielles basées sur Java ES, notamment des analystes d'exploitation, des architectes système, des ingénieurs de projet et des administrateurs système.

Les lecteurs de ce manuel 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
- Langage de programmation Java
- Composants Java 2 Platform, Standard Edition (J2SE™ platform) et Java 2 Platform, Enterprise Edition (J2EE™ platform)

## Organisation de ce document

Ce manuel comprend les chapitres suivants :

- Le [Chapitre 1](#) présente Java ES et les tâches impliquées dans l'utilisation du système.
- Le [Chapitre 2](#) décrit une structure d'architecture pour la conception d'architectures de solutions Java ES et fournit un exemple d'architecture basée sur cette structure.
- Le [Chapitre 3](#) fournit des informations sur les fonctions jouant un rôle clé dans l'intégration de composants Java ES dans un système logiciel unique.
- Le [Chapitre 4](#) décrit les concepts et la terminologie propres à chaque phase du cycle de vie d'une solution Java ES.

- L'[Annexe A](#) répertorie les composants Java ES.

## Documentation Logiciel Java Enterprise System

La documentation relative à Java ES décrit la planification du déploiement et l'installation système. L'URL de la documentation système est <http://docs.sun.com/coll/1286.3>. Pour une présentation de Java ES, reportez-vous aux manuels dans l'ordre dans lequel ils apparaissent dans le tableau ci-dessous. De nombreuses informations et ressources sur Java ES sont également disponibles à l'adresse <http://www.sun.com/bigadmin/hubs/javaes/>.

TABLEAU P-1 Documentation de Logiciel Java Enterprise System

Titre du document	Sommaire
<i>Sun Java Enterprise System 5 Update 1 Release Notes</i>	Contient les informations les plus récentes concernant Java ES, y compris les problèmes connus. En outre, les composants possèdent leurs propres notes de version dans la collection de notes de version ( <a href="http://docs.sun.com/coll/1315.3">http://docs.sun.com/coll/1315.3</a> ).
Sun Java Enterprise System 5 Update 1 - Nouveautés	Présente un récapitulatif des nouvelles fonctionnalités de Java ES 5 Update 1. Propose des liens vers la documentation mise à jour.
<i>Présentation technique de Sun Java Enterprise System 5 Update 1</i>	Décrit les principes techniques et conceptuels de Java ES. Décrit les composants, l'architecture, les processus et les fonctions.
<i>Sun Java Enterprise System Deployment Planning Guide</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>Sun Java Enterprise System 5 Installation Planning Guide</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>Sun Java Enterprise System 5 Update 1 Installation Guide for UNIX</i>	Vous guide tout au long de la procédure d'installation de Java ES. Il vous explique comment configurer les composants après leur installation et vous assurer de leur fonctionnement correct.
<i>Sun Java Enterprise System 5 Installation Guide for Microsoft Windows</i>	

TABLEAU P-1 Documentation de Logiciel Java Enterprise System (Suite)

Titre du document	Sommaire
<i>Référence de l'installation de Sun Java Enterprise System 5 Update 1 pour UNIX</i>	Fournit des informations supplémentaires sur les paramètres de configuration et des feuilles de travail à utiliser dans la planification de votre configuration et répertorie les éléments de référence comme les répertoires par défaut et les numéros de port sous Solaris et dans l'environnement d'exploitation Linux.
<i>Sun Java Enterprise System 5 Update 1 Upgrade Guide for UNIX</i>	Fournit des instructions sur la mise à niveau vers Java ES 5 Update 1 à partir des versions précédemment installées.
<i>Sun Java Enterprise System 5 Update 1 Upgrade Guide for Microsoft Windows</i>	
<i>Sun Java Enterprise System 5 Update 1 Monitoring Guide</i>	Fournit des instructions de configuration de Monitoring Framework pour chaque composant produit et utilise Monitoring Console pour afficher les données en temps réel et créer des règles de surveillance.
<i>Sun Java Enterprise System Glossary</i>	Définit les termes utilisés dans la documentation de Java ES.

## Conventions typographiques

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

TABLEAU P-2 Conventions typographiques

Type de caractères	Signification	Exemple
AaBbCc123	Nom des commandes, fichiers et répertoires, ainsi que messages système.	Modifiez votre fichier <code>.login</code> . Utilisez <code>ls -a</code> pour afficher la liste de tous les fichiers. <code>nom_machine%</code> Vous avez reçu du courrier.
<b>AaBbCc123</b>	Ce que vous entrez, par opposition à ce qui s'affiche à l'écran.	<code>nom_machine% su</code> Mot de passe :
<i>AaBbCc123</i>	Marque de réservation à remplacer par une valeur ou un nom réel.	La commande de suppression d'un fichier est <code>rm nom-de-fichier</code> .
<i>AaBbCc123</i>	Titres de manuels, nouveaux termes et termes à mettre en valeur (certains éléments mis en valeur sont affichés en gras)	Reportez-vous au chapitre 6 du <i>Guide de l'utilisateur</i> . Un <i>cache</i> est une copie des éléments stockés localement. <i>N'enregistrez pas</i> le fichier.

## Documentation, support et formation

Le site Web Sun fournit des informations sur les ressources supplémentaires suivantes :

- [Documentation \(http://www.sun.com/documentation/\)](http://www.sun.com/documentation/)
- [Support \(http://www.sun.com/support/\)](http://www.sun.com/support/)
- [Formation \(http://www.sun.com/training/\)](http://www.sun.com/training/)

## Recherche de documentation sur les produits Sun

Vous pouvez rechercher les documentations des produits Sun à partir du site Web docs.sun.com<sup>SM</sup>, mais vous pouvez également utiliser un moteur de recherche en tapant la syntaxe suivante dans le champ de recherche :

*Terme-de-recherche* site:docs.sun.com

Par exemple, pour rechercher "broker", saisissez ce qui suit :

broker site:docs.sun.com

Pour inclure d'autres sites Web Sun dans votre recherche (par exemple, [java.sun.com](http://java.sun.com), [www.sun.com](http://www.sun.com) et [developers.sun.com](http://developers.sun.com)), utilisez sun . com au lieu de docs . sun . com dans le champ de recherche.

## Références à des sites Web tiers

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

---

**Remarque** – Sun ne peut être tenu responsable de la disponibilité des sites Web des tiers mentionnés dans le présent document. Sun ne garantit pas le contenu, la publicité, les produits et autres documents 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 820-2969.



# 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. Ce manuel traite essentiellement des composants logiciels de Java ES et des services qu'ils fournissent.

Ce chapitre présente Java ES et les tâches qu'implique l'utilisation du système. Le chapitre se compose des sections suivantes :

- “À quoi sert Java Enterprise System ?” à la page 17
- “Composants Java ES” à la page 19
- “Composants des Suites Sun Java” à la page 24
- “Utilisation de Java ES” à la page 26
- “Termes clés de ce chapitre” à la page 31

## 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 ES 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 ES fournit la couche de services d'infrastructure distribuée présentée dans la figure.

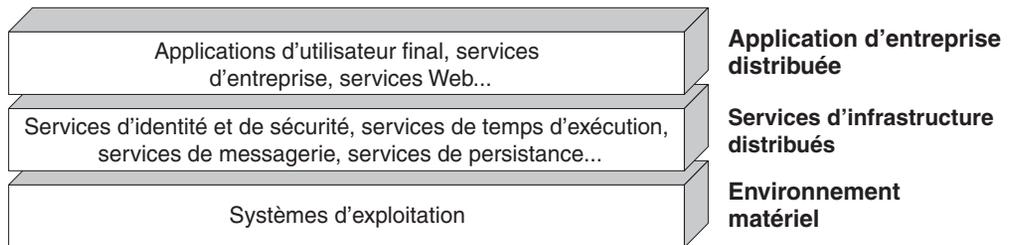


FIGURE 1-1 Prise en charge requise pour les applications d'entreprise distribuées

Parmi les services fournis par Java ES figurent :

- **Services de portail** : ces services permettent aux employés, télétravailleurs, experts, partenaires commerciaux, fournisseurs et clients d'accéder aux ressources de l'entreprise quel 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 communication et de collaboration** : ces services permettent un échange d'informations sécurisé entre les diverses communautés d'utilisateurs. Parmi les fonctionnalités spécifiques, il faut citer la messagerie, la collaboration en temps réel (messagerie instantanée, conférences) ainsi que la planification du travail de l'utilisateur.

---

**Remarque** – Ce manuel fait référence à des composants de Sun Java Communications Suite, qui dépendent des composants Java ES et sont utilisés au sein d'architectures de déploiement Java ES. Les composants de communication et de collaboration ne sont pas inclus dans Java ES.

---

- **Services réseau d'identité et de sécurité** : 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 technologie J2EE.

Java ES 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 ES, 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 ES.
- **Services de contrôle** : ces services fournissent des informations en temps réel sur les composants Java ES.

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

## Composants Java ES

Java ES est une intégration de composants et produits logiciels discrets sous la forme d'un système logiciel unique. Leur intégration est facilitée par un certain nombre de fonctions au niveau du système, répertoriées ci-après :

- Tous les composants sont synchronisés sur un ensemble commun de bibliothèques 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.

- Tous les composants Java ES disposent d'une structure de contrôle commune.

Ces fonctions sont expliquées dans les chapitres suivants de ce manuel. Cette section a pour objet de présenter les divers composants intégrés à Java ES. Ces **composants système** peuvent être regroupés en trois catégories principales, comme indiqué dans l'illustration suivante :

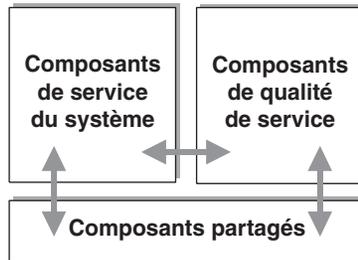


FIGURE 1-2 Catégories de composants Java ES

Les composants fournissent les services suivants :

- **Composants de services système** : ils fournissent les services d'infrastructure principaux de Java ES 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.

Pour obtenir la liste des composants Java ES, reportez-vous à l'[Annexe A](#).

## Composants de services 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 d'identité et de sécurité, les services de conteneur Web, les services d'application J2EE ainsi que les services de persistance.

Les **composants de services 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 à unités d'exécution multiples capable de prendre en charge un nombre important de clients. Pour plus de détails sur chaque composant, reportez-vous à la section "**Composants de services système**" à la page 71

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 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.
Java DB <sup>1</sup>	Fournit une base de données légère pour le développement d'applications Java. Java DB est la distribution prise en charge par Sun de la base de données de technologie open-source Apache Derby 100% Java.
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.
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 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 des schémas XML, des règles des processus industriels, du contrôle d'accès, du contrôle de version, etc.

<sup>1</sup> Java ES 5 est la première version à inclure Java DB comme composant logiciel. Dans sa première version, Java DB était un composant partagé appelé Base de données Derby inclus dans Java ES 2005Q4.

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

Composant	Services système fournis
Sun Java System Web Server	Fournit des services de conteneur Web J2EE pour les composants Web Java tels que les composants Java™ servlet et JavaServer Pages™ (JSP™). Web Server prend également en charge d'autres technologies d'application Web pour la fourniture de contenus Web statiques et dynamiques, tels que les scripts CGI et Sun Java System Active Server Pages.

## Composants de qualité de service

Outre les composants de service système présentés dans le [Tableau 1-1](#), Java ES 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 de contrôle

## 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 ES, ainsi que les services qu'ils fournissent. Pour plus de détails sur chaque composant, reportez-vous à la section [“Composants de disponibilité”](#) à la page 76.

TABLEAU 1-2 Composants de disponibilité de Java ES

Composant	Services de disponibilité fournis
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.
Sun Cluster	Fournit des services de disponibilité et d'évolutivité élevées pour Java ES, les applications exécutées en haut de l'infrastructure Java ES et l'environnement matériel dans lequel ces deux entités sont déployées.
Sun Cluster Geographic Edition <sup>1</sup>	Ce composant protège les applications contre les arrêts brutaux, à l'aide de plusieurs clusters répartis sur plusieurs emplacements géographiques et d'une infrastructure redondante répliquant les données entre ces clusters. Le logiciel Sun Cluster Geographic Edition est une extension en couches du logiciel Sun Cluster.

<sup>1</sup> Java ES 5 est la première version à inclure Sun Cluster Geographic Edition en tant que composant logiciel Java ES.

## 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 et de mise en cache. Le tableau suivant présente les composants d'accès inclus dans Java ES, ainsi que les services qu'ils fournissent. Pour plus de détails sur chaque composant, reportez-vous à la section “Composants d'accès” à la page 78.

TABLEAU 1-3 Composants d'accès de Java ES

Composant	Services d'accès fournis
Sun Java System Portal Server (comprend 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 Web Proxy Server	Assure la mise en cache, le filtrage et la distribution du contenu Web pour les requêtes Internet entrantes et sortantes.

## Composants de contrôle

Java ES comprend une fonction de contrôle fournissant l'état du système en temps réel et des tâches de contrôle personnalisables. Le contrôle est implémenté par Sun Java System Monitoring Console [composant de produit](#), pris en charge par Sun Java System Monitoring Framework [shared component \(composant partagé\)](#). Pour plus d'informations, consultez la section “Composants de contrôle” à la page 79.

## Composants partagés

Java ES comprend un certain nombre de bibliothèques partagées installées localement, dont dépendent de nombreux composants de services système et de qualité de service. Les [composants partagés](#) Java ES fournissent des services locaux aux [composants de produit](#) 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 Java ES : Java 2 Platform, Standard Edition (J2SE), Netscape Portable Runtime (NSPR), Network Security Services (NSS), Java Security Services for Java (JSS), etc. Pour la liste complète, reportez-vous à la section “Composants partagés” à la page 79.

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.

## Composants des Suites Sun Java

Java ES est disponible à la fois en tant que distribution logicielle unique d'infrastructures de bout en bout et en tant que distributions individuelles de suites ciblant les besoins spécifiques critiques des entreprises. Java ES comprend tous les composants Java ES, tandis que les suites Sun Java System comprennent des sous-ensembles sélectionnés pour répondre aux besoins spécifiques des entreprises. Les programmes d'installation et de désinstallation sont compris dans toutes les distributions de suites, mais sont allégés pour gérer uniquement les composants de la suite. Tous les composants partagés sont également inclus dans les distributions de suite.

Le tableau suivant répertorie le contenu des suites individuelles et les exigences de l'entreprise que chacune doit satisfaire.

TABLEAU 1-4 Composants des Suites Sun Java

Suite	Exigence de l'entreprise	Sommaire
Sun Java Application Platform Suite	Développer, déployer et gérer des architectures SOA (Service-Oriented Architectures) de prochaine génération	Access Manager Application Server Directory Server HADB Java DB Message Queue Console de contrôle Portal Server (comprend Secure Remote Access et Mobile Access) Service Registry Serveur Web Proxy Web Server
Sun Java Availability Suite	Reprise sur sinistre et haute disponibilité pour applications critiques	Logiciel Sun Cluster Agents Sun Cluster Édition géographique de Sun Cluster

TABLEAU 1-4 Composants des Suites Sun Java (Suite)

Suite	Exigence de l'entreprise	Sommaire
Sun Java Communications Suite <sup>1</sup>	Services de messagerie et de collaboration sécurisés et fiables	Access Manager Application Server Calendar Server* Communications Express* Delegated Administrator* Directory Server HADB Instant Messaging* Java DB Message Queue Messaging Server* Console de contrôle Serveur Web Proxy Web Server
Sun Java Identity Management Suite	Gestion des identités utilisateur entre plusieurs infrastructures et environnements d'application informatiques.	Access Manager Application Server Directory Server HADB Java DB Message Queue Console de contrôle Web Server

<sup>1</sup> Les composants suivis d'un astérisque (\*) sont des composants de communication qui ne sont plus ni fournis avec Java ES ni installés avec le programme d'installation de Java ES. Ces composants sont disponibles en tant que composants de Sun Java Communications Suite.

TABLEAU 1-4 Composants des Suites Sun Java (Suite)

Suite	Exigence de l'entreprise	Sommaire
Sun Java Web Infrastructure Suite	Applications et services Web pour les petites à moyennes entreprises.	Access Manager Application Server Directory Server HADB Java DB Message Queue Console de contrôle Service Registry Serveur Web Proxy Web Server

## Utilisation de Java ES

La création de solutions d'entreprise basées sur le logiciel Java ES implique plusieurs tâches standard. L'étendue et la complexité de ces tâches varient selon le point de départ de l'adoption de Java ES et la nature de la solution que vous tentez de créer et de déployer.

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

## Cycle de vie d'une solution Java ES

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 suivante. Cette figure présente également la catégorie d'utilisateurs de Java ES qui effectue les diverses tâches en règle générale.

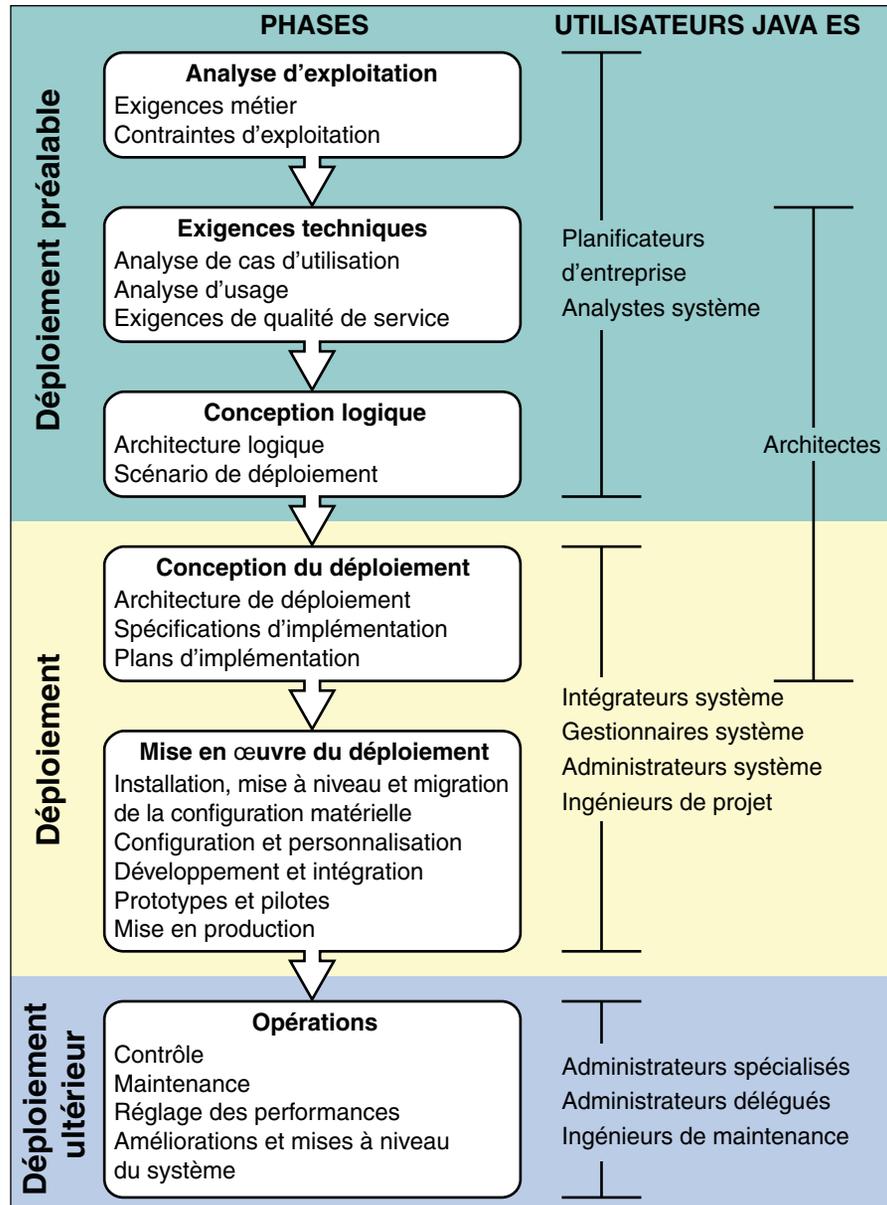


FIGURE 1-3 Phases du cycle de vie d'une solution et catégories d'utilisateurs

Les phases du cycle de vie indiquées dans la figure précédente peuvent être divisées en trois grands groupes :

- **Prédéploiement** : au cours de ces phases, une exigence de l'entreprise est convertie en un scénario de déploiement, qui est une architecture logique et un ensemble d'exigences 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.
- **Postdéploiement** : 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.

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. 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. Connait 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. Connait les technologies les plus récentes. Comprend les exigences et les contraintes d'entreprise.	Exigences techniques Conception logique Conception de 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 de déploiement
Ingénieur de projet	Connait 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 technique	Connait les protocoles, les périphériques, la sécurité et l'architecture réseau. Connait les langages de script et de programmation.	
Administrateur système spécialisé	Possède des connaissances sur le produit ou des connaissances techniques spécialisées.	Opérations
Administrateur délégué	Connait 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. Connait l'administration système pour les plates-formes du système d'exploitation.	

## Scénarios d'adoption de Java ES

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 vous ne disposez encore d'aucun système logiciel, vous pouvez déployer le logiciel Java ES 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.
- **Extension** : si vous disposez déjà d'une infrastructure informatique, vous pouvez déployer le logiciel Java ES 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 le groupement personnalisé 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 Logiciel Java Enterprise System ou de produits Sun antérieurs à Java ES, vous effectuez une mise à niveau pour obtenir la dernière version des composants Java ES.

Chaque scénario d'adoption contient ses propres considérations et défis. Toutefois, selon le scénario d'adoption utilisé, les problèmes à traiter et les ressources requises pour investir dans les phases du cycle de vie présentées dans la [Figure 1-3](#) peuvent varier.

Les considérations suivantes 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 ES.
- **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	Peut être un gros problème	Compromis entre les coûts de l'équipement et de main d'œuvre. <sup>1</sup>

<sup>1</sup> 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.

TABLEAU 1-6 Problèmes relatifs aux scénarios d'adoption de Java ES (Suite)

Scénario d'adoption	Migration	Intégration	Formation	Matériel
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
Mettre à niveau	Peut être un gros problème	Intégration des composants mis à niveau relativement facile	Problème relativement mineur	Problème relativement mineur

## 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 ES.

<b>adoption scenario (scénario d'adoption)</b>	Raison principale justifiant le déploiement du logiciel Java ES et caractérisant le système logiciel de départ et l'objectif que vous cherchez à atteindre. Il existe principalement quatre scénarios d'adoption de Java ES : nouveau système, remplacement, extension et mise à niveau.
<b>composant</b>	Unité de logique logicielle à partir de laquelle sont élaborées les applications distribuées. Un composant peut être l'un des <a href="#">composants système</a> inclus dans Java ES ou un <a href="#">application component (composant d'application)</a> 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 <a href="#">services d'entreprise</a> et peuvent être encapsulés sous la forme de <a href="#">services Web</a> .
<b>distributed enterprise application (application d'entreprise distribuée)</b>	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).
<b>end user (utilisateur final)</b>	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' <a href="#">deployment architecture (architecture de déploiement)</a> de l'application.

<b>service</b>	Fonction logicielle exécutée pour un ou plusieurs <a href="#">clients</a> . 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 <a href="#">business service (service d'entreprise)</a> 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).
<b>composant de produit</b>	<a href="#">Composants de services système</a> Java ES, fournissant les principaux services d'infrastructure ES et <a href="#">Composants de qualité de service</a> Java ES, qui les améliorent. Les composants de produit sont programmables au sein du programme d'installation de Java ES.
<b>service quality component (composant de qualité de service)</b>	Type de <a href="#">system component (composant système)</a> inclus dans Java ES. 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.
<b>shared component (composant partagé)</b>	Type de <a href="#">system component (composant système)</a> inclus dans Java ES. Les composants partagés, en règle générale des bibliothèques, fournissent des services locaux aux autres composants du système.
<b>system component (composant système)</b>	Tout package logiciel ou ensemble de packages inclus dans Java ES et installé par le programme d'installation de Java ES. Il existe plusieurs types de composants système : les <a href="#">composants de produit</a> qui fournissent les services d'infrastructure Java ES et les <a href="#">composants partagés</a> qui fournissent des services locaux aux autres composants système.
<b>service système</b>	Un ou plusieurs <a href="#">services</a> distribués qui définissent la fonctionnalité unique offerte par Java ES. Les services système exigent normalement la prise en charge d'un certain nombre de <a href="#">composants de qualité de service</a> , de <a href="#">composants partagés</a> ou les deux.
<b>system service component (composant de services système)</b>	Type de <a href="#">system component (composant système)</a> inclus dans Java ES. Les composants de services de Java ES fournissent les principaux services d'infrastructure : services de portail, d'identité et de sécurité, Web et d'application et de disponibilité.

## Architectures de solutions Java ES

---

Ce chapitre fournit un aperçu des concepts architecturaux sur lesquels sont basées les solutions Java ES. Il explique comment les composants de services système et 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 [logical architecture \(architecture logique\)](#) et une [deployment architecture \(architecture de déploiement\)](#). 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, et un exemple d'architecture s'appuyant sur cette structure. Le chapitre se compose des sections suivantes :

- [“Structure architecturale Java ES” à la page 33](#)
- [“Exemple d'architecture de solution Java ES” à la page 47](#)
- [“Termes clés de ce chapitre” à la page 50](#)

### Structure architecturale Java ES

Les composants Java ES prennent en charge le déploiement de solutions logicielles 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 d'entreprise. 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 souligne l'importance du rôle des composants de services système dans la prise en charge de solutions distribuées (voir la section “Composants de services système” à la page 20).
- **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 22).

Ces trois dimensions sont illustrées dans la figure suivante.

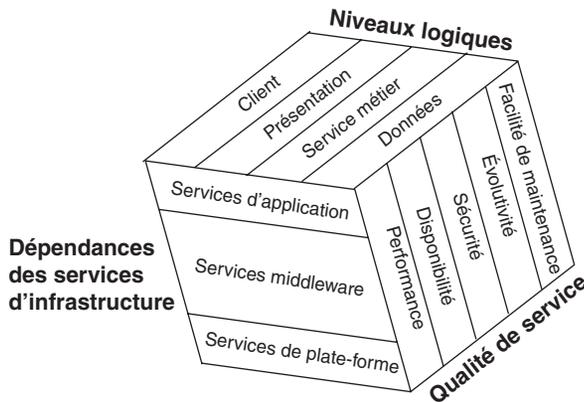


FIGURE 2-1 Dimensions de l'architecture d'une solution Java ES

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

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

## Dimension 1 : dépendances des services d'infrastructure

Les composants logiciels en interaction des applications d'entreprise distribuées nécessitent des services d'infrastructure sous-jacents, 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.

Les dépendances des services d'infrastructure de l'architecture d'une solution sont illustrées 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 constituent la principale dimension de ces composants de services système Java ES (voir la section “[Composants de services système](#)” à la page 20).

En général, les services présentés dans la figure ci-après 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.

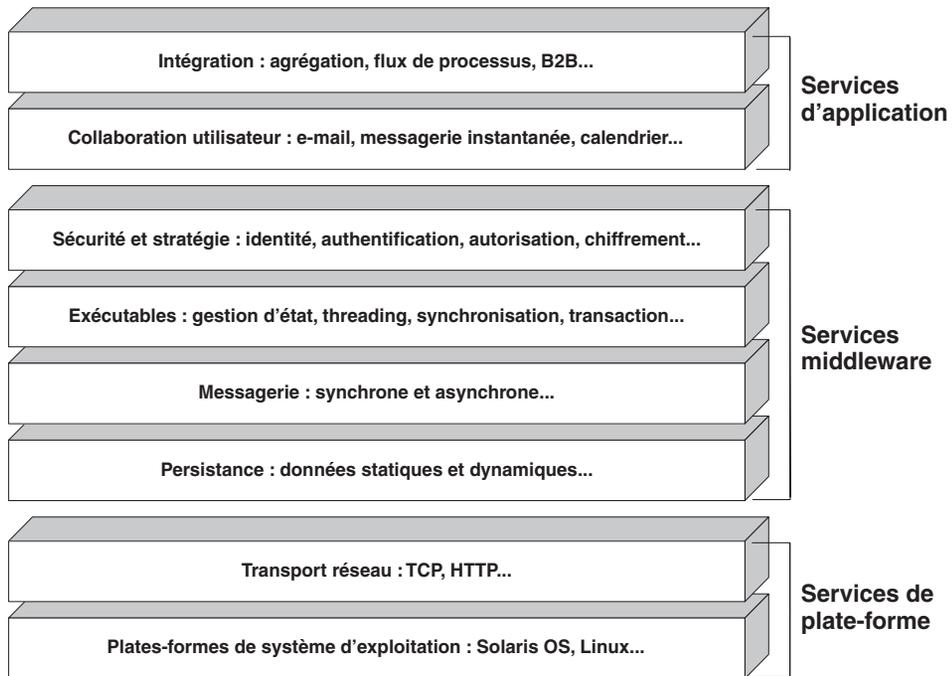


FIGURE 2-2 Dimension 1 : niveaux des services d'infrastructure

Les descriptions suivantes des différents niveaux de services d'infrastructure font référence aux artefacts du langage de programmation Java, le cas échéant ; ils sont répertoriés 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 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.
- **Persistance** : 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'application. 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 les accès basés sur les 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 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 travaux. 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 des 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 des services d'architecture Java ES

Les composants de Java ES implémentent les niveaux de services d'infrastructure distribués illustrés à la [Figure 2-2](#). Le positionnement des composants de services système au sein des différents niveaux est illustré dans la figure ci-après.

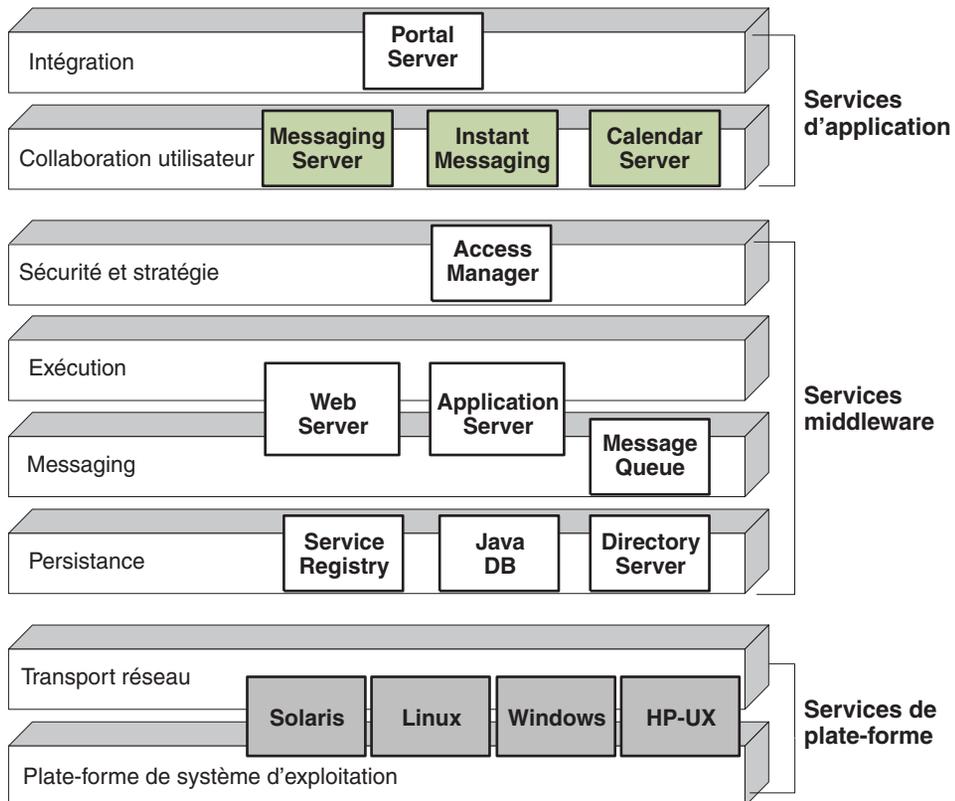


FIGURE 2-3 Composants de service système Java ES

**Remarque** – Les boîtes grisées dans la figure signalent des composants qui ne sont pas inclus dans Java ES. Les composants de collaboration utilisateur ne font pas partie de Java ES mais sont souvent déployés avec les composants Java ES et utilisés au sein d'architectures Java ES. Ces composants font partie de Sun Java Communications Suite et sont cités dans ce document à des fins d'illustration uniquement. Par ailleurs, les plates-formes de systèmes d'exploitation ne font pas véritablement partie de Java ES ; toutefois, elles ont été incluses afin de montrer les plates-formes de systèmes d'exploitation sur lesquelles les composants Java ES sont pris en charge.

## Dépendances des services d'infrastructure Java ES

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 ci-après montre les relations spécifiques entre les composants de services 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 et Instant Messaging <sup>1</sup>	Aucun
Access Manager	Application Server ou Web Server Directory Server	Portal Server Si configuré pour une connexion unique : Calendar Server, Messaging Server et 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
Web Server	Access Manager (pour le contrôle d'accès)	Portal Server Access Manager
Directory Server	Aucun	Portal Server Access Manager Calendar Server Messaging Server Instant Messaging
Service Registry	Java DB	Composants basés sur Application Server
Java DB	Aucun	Service Registry

<sup>1</sup> Les composants Calendar Server, Messaging Server et Instant Messaging sont disponibles dans Sun Java Communications Suite.

## 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.

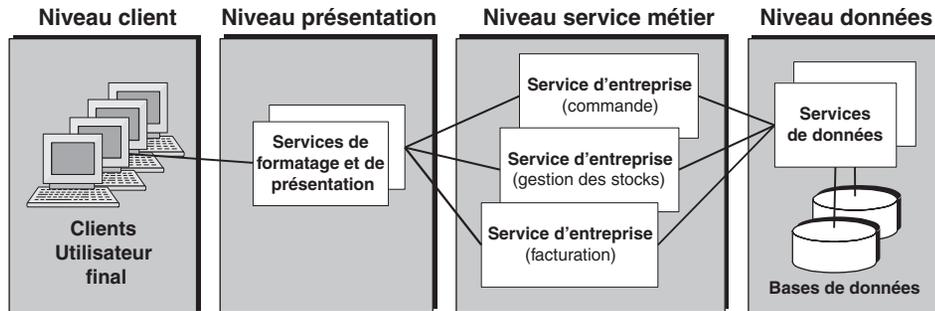


FIGURE 2-4 Dimension 2 : niveaux logiques pour les applications d'entreprise distribuées

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 35 assurent la prise en charge des composants d'application dans tous les niveaux logiques illustrés dans la Figure 2-4. Si les concepts de niveau logique s'appliquent essentiellement aux applications d'entreprise personnalisées, ils s'appliquent également aux services de collaboration fournis par les composants Sun Java Communications Suite et certains services de portail.

### Description des niveaux logiques

Cette section fournit une brève description des quatre niveaux logiques représentés dans la Figure 2-4. Ces descriptions concernent les composants d'application implémentés à l'aide du modèle de composant de la 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 (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 services 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). 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 répllication 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. Certaines de ces solutions englobent des services de niveau application fournis directement par les composants Sun Java Communications Suite et certains composants 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 dans la figure suivante, où les lignes reliant les composants représentent les interactions.

---

**Remarque** – La figure suivante ne représente pas une architecture logique complète. Un certain nombre de composants Java ES ont été omis afin de simplifier le schéma.

---

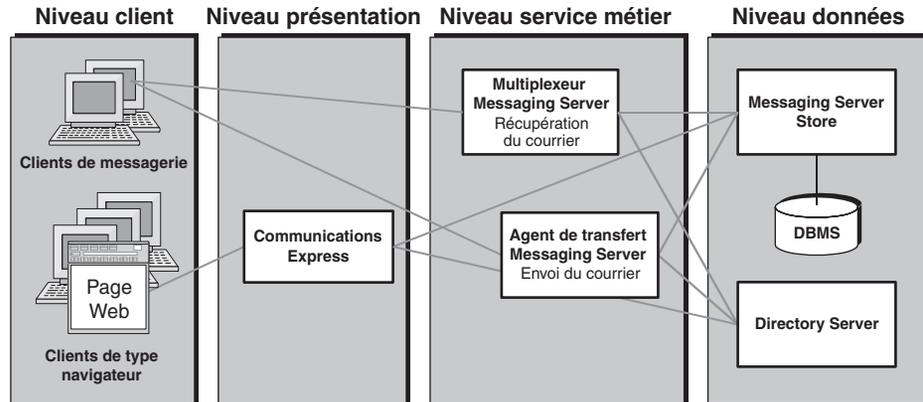


FIGURE 2-5 Messaging Server : exemple d'architecture à plusieurs niveaux

**Remarque** – Les composants de communication ne font pas partie de Java ES mais sont souvent déployés avec les composants Java ES et utilisés au sein d'architectures Java ES. Ces composants de communication font partie de Sun Java Communications Suite et sont cités dans ce document à des fins d'illustration uniquement.

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 43](#)). 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. 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' <i>activité</i> d'un système).
Sécurité	Combinaison complexe de facteurs décrivant l'intégrité d'un système et de ses utilisateurs. La sécurité implique la sécurité physique des systèmes, la sécurité du réseau, la sécurité des applications et des données (authentification et autorisation des utilisateurs) ainsi que le transport sécurisé des informations.
Évolutivité	Possibilité d'ajouter de la capacité à un système déployé dans le temps. En principe, l'évolutivité implique l'ajout de ressources au système, mais ne doit pas entraîner de changements au niveau de l'architecture du déploiement.
Capacité latente	Aptitude d'un système à traiter une utilisation de charge de pointe inhabituelle sans ressources supplémentaires.
Entretien	Facilité avec laquelle un système déployé peut être entretenu. Cela inclut le contrôle du système, la résolution des problèmes 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'influer 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 Java ES

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-feu.

Les composants de qualité de service de Java ES, présentés dans la section “[Composants de qualité de service](#)” à la page 22, sont récapitulés ci-après :

- **Composants de disponibilité** : assurent un temps d'activité quasi continu d'une solution déployée.
- **Composants d'accès** : fournissent un accès Internet sécurisé aux services système ainsi qu'une fonction de routage.
- **Composants de contrôle** : fournissent des informations en temps réel sur les composants Java ES.

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
High Availability Session Store	Disponibilité
Console de contrôle	Entretien
Portal Server Secure Remote Access	Sécurité Extensibilité
Sun Cluster	Disponibilité Extensibilité
Édition géographique de Sun Cluster	Disponibilité Extensibilité
Serveur Web Proxy	Sécurité Performances Extensibilité

## 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 de panne, 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.

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.

Une extension au logiciel Sun Cluster software est fournie par Sun Cluster Geographic Edition, qui protège les applications contre les arrêts brutaux à l'aide de plusieurs clusters répartis dans des emplacements géographiques différents et d'une infrastructure assurant la réplication des données entre les clusters.

---

**Remarque** – Sun Cluster et Sun Cluster Geographic Edition sont pris en charge sur le système d'exploitation Solaris™ (SE Solaris) uniquement.

---

## 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 Java ES. Une panne de Directory Server pouvant avoir un impact énorme sur un 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.

Ce manuel ne détaille pas les différentes méthodologies de conception basées sur la structure architecturale représentée par la section "[Structure architecturale Java ES](#)" à la [page 33](#). Toutefois, la structure d'architecture tridimensionnelle souligne les aspects de conception importants pour comprendre le déploiement des solutions logicielles basées sur Logiciel Java Enterprise System.

## Exemple d'architecture de solution Java ES

Java ES 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 ES. 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 encapsuler 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 ES 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 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	Points d'accès Web uniques et personnalisés aux services de communication, tels que la messagerie électronique, 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.

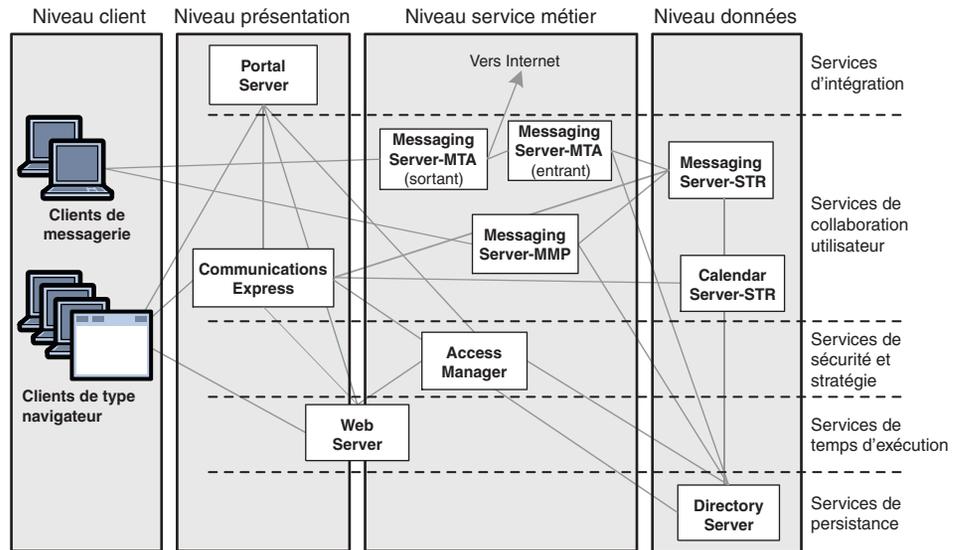


FIGURE 2-6 Architecture logique d'un scénario de communication d'entreprise

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 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 *Sun Java Enterprise System Deployment Planning Guide*.

## 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-éseaux 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 multimaitre peut être utilisée pour Directory Server.

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

## 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 ES.

### application component (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 **distributed enterprise application** (**application d'entreprise distribuée**), la conception architecturale inclut généralement à la fois l'**logical architecture** (**architecture logique**) de l'application et l'**deployment architecture** (**architecture de déploiement**).

### business service (service d'entreprise)

**application component** (**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 à unités d'exécution multiples). Un service d'entreprise peut également être un assemblage de composants encapsulés sous la forme d'un **web service** (**service Web**), ou d'un **serveur** autonome.

### client

Logiciel demandant des **services** logiciels. Un client peut être un service qui requiert un autre service ou un composant de l'interface graphique auquel accède l'utilisateur final.

---

<b>deployment architecture (architecture de déploiement)</b>	Conception de haut niveau décrivant le mappage d'une <b>logical architecture (architecture logique)</b> 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 les logiciels.
<b>logical architecture (architecture logique)</b>	Conception décrivant les blocs constitutifs d'une application distribuée et les relations (ou interfaces) entre ces blocs. L'architecture logique comprend aussi bien les <b>composants d'application</b> distribués que les services d'infrastructure requis pour leur prise en charge.
<b>serveur</b>	Processus logiciel à unités d'exécution multiples (différent d'un serveur matériel) fournissant un <b>service</b> distribué ou un ensemble cohérent de services pour des <b>clients</b> accédant au service par le biais d'une interface externe.
<b>web service (service Web)</b>	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 ES

---

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 ES par rapport à l'intégration manuelle de différents produits d'infrastructure.

Le chapitre se compose des sections suivantes :

- “Programme d'installation intégré de Java ES” à la page 53
- “Services de contrôle système” à la page 55
- “Services intégrés d'identité et de sécurité” à la page 55
- “Termes clés de ce chapitre” à la page 59

### Programme d'installation intégré de Java ES

Tous les composants de Java ES sont installés à l'aide d'un programme d'installation unique. 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 hôte 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 les langues suivantes : le français, l'allemand, le japonais, le coréen, l'espagnol, le chinois simplifié et le chinois traditionnel.

Cette section aborde les différents aspects du programme d'installation intégré de Java ES . Pour plus d'informations, reportez-vous au *Sun Java Enterprise System 5 Update 1 Installation Guide for UNIX*.

## Contrôle de logiciels préexistants

Le programme d'installation examine l'hôte 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 interopérabilité. 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 comme J2SE ou NSS, sont déjà installés et répertorie les incompatibilités (voir [“Composants partagés” à la page 23](#)). 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 fonctionnent bien conjointement. La plupart des composants ont des dépendances par rapport aux autres composants. 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 sert à 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 ES fournit également un programme de désinstallation permettant de 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 de contrôle système

Java ES comprend une nouvelle fonctionnalité de contrôle permettant de contrôler en temps réel les services système. Le contrôle est implémenté par Sun Java System Monitoring Framework ([shared component \(composant partagé\)](#)) et Sun Java System Monitoring Console ([composant de produit](#)). Monitoring Framework est automatiquement configuré et activé pour collecter des données pour chaque composant Java ES installé et Monitoring Console est l'interface graphique utilisée pour afficher les données contrôlées. Monitoring Console est un composant pouvant être sélectionné au cours de l'installation de Java ES tandis que Monitoring Framework est installé automatiquement.

Le contrôle est le processus consistant à collecter des données d'exécution et à calculer la qualité des critères de service afin que les administrateurs système puissent évaluer les performances et recevoir des alarmes. Au cours de l'exécution, les administrateurs sont en interaction avec Monitoring Console pour afficher les statistiques de performance, définir les seuils à surveiller de manière dynamique, définir des tâches de contrôle personnalisées et accuser réception des alarmes.

## Services intégrés d'identité et de sécurité

L'une des fonctions importantes de Java ES est la gestion intégrée des identités des utilisateurs et la structure d'autorisation et d'authentification intégrée. Cette section fournit des informations techniques permettant de comprendre les services intégrés d'identité et de sécurité proposés par Java ES.

## Identité unique

Au sein d'un environnement Java ES, un utilisateur final possède une seule identité intégrée. En fonction de cette **single identity (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 **directory (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 ES, 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.

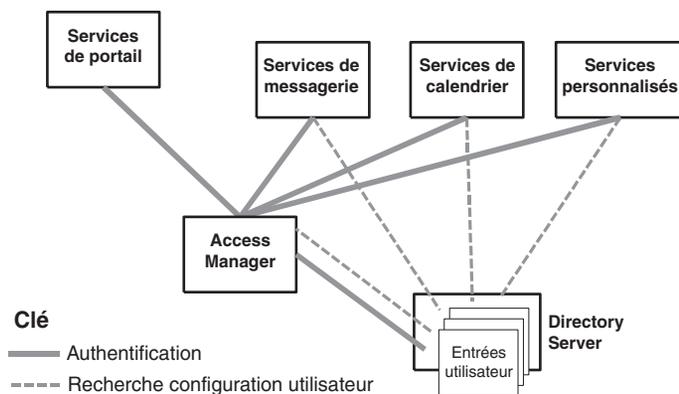


FIGURE 3-1 Une entrée utilisateur unique prend en charge de nombreux services

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 ES.

## 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 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, les fonctionnalités de l'agent de stratégie sont assurées par un sous-composant Access Manager SDK.

### 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.

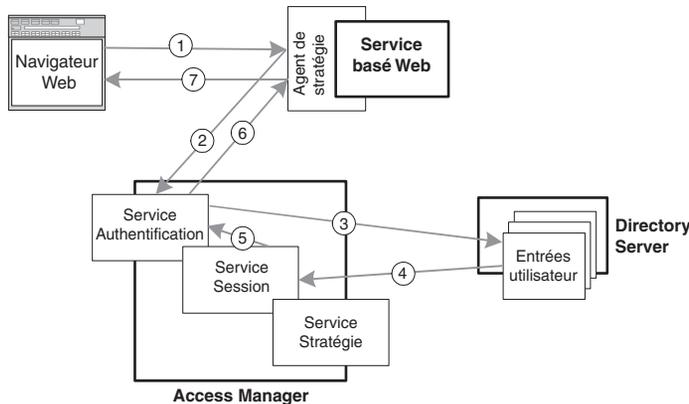


FIGURE 3-2 Séquence d'authentification

## 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 ES. 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 **policy (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.

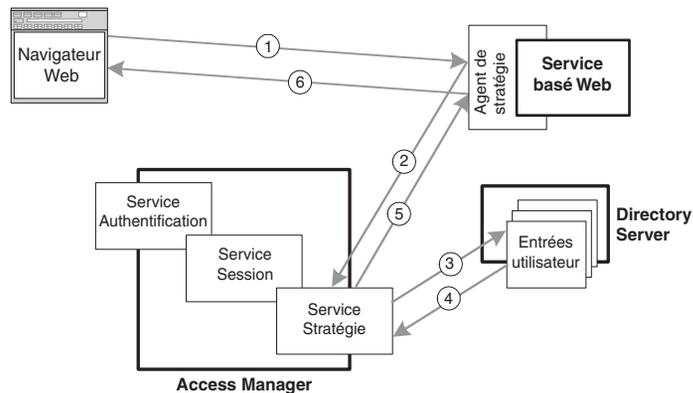


FIGURE 3-3 Séquence d'autorisation

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 ES.

**directory**  
(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).

**policy**  
(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.

single identity  
(identité unique)

---

**single identity  
(identité unique)**

Identité affectée à un utilisateur par le biais d'une entrée utilisateur unique dans un annuaire Java ES. Sur la base de cette entrée utilisateur unique, l'utilisateur peut obtenir l'accès à diverses ressources Java ES, telles que portails, pages Web et services (comme la messagerie, le calendrier ou la messagerie instantanée).

**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.

# Cycle de vie d'une solution Java ES

---

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 décrit les tâches intervenant à chaque phase du cycle de vie. Le chapitre se compose des sections suivantes :

- “Prédéploiement” à la page 63
- “Déploiement” à la page 64
- “Postdéploiement” à la page 69
- “Termes clés de ce chapitre” à la page 69

## 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. Le diagramme du cycle de vie est repris pour faciliter les références.

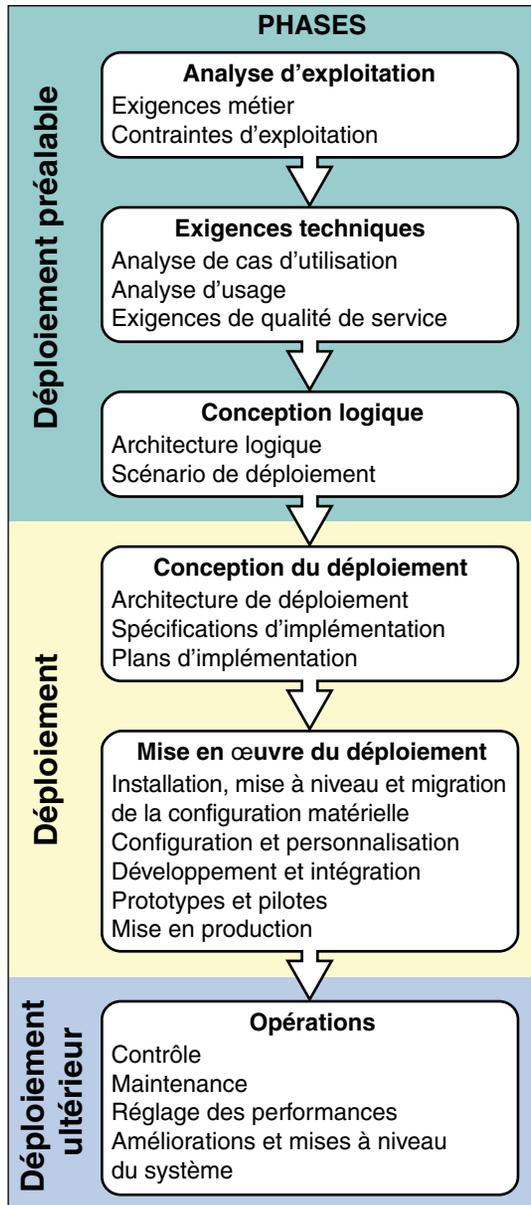


FIGURE 4-1 Tâches du cycle de vie d'une solution

# Prédéploiement

Au cours des phases de **prédéploiement** du cycle de vie, vous transformez une analyse des besoins d'entreprise en un **deployment scenario (scénario de déploiement)**. Le scénario de déploiement fait office de spécification pour une conception de déploiement.

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

- **Analyse d'exploitation** : 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** : 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** : 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 prédéploiement du cycle de vie, reportez-vous au *Sun Java Enterprise System Deployment Planning Guide*.

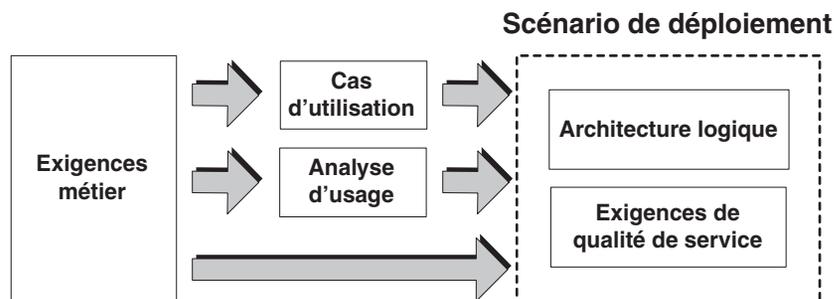


FIGURE 4-2 Spécification d'un scénario de déploiement

# 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 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 de déploiement”](#) à la page 64. 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 66. l'implémentation d'une conception de déploiement est un processus 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 ces deux phases du processus de déploiement.

## Conception de 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.

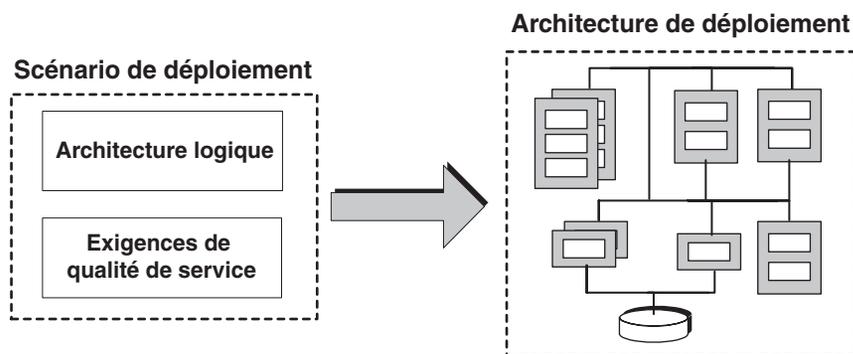


FIGURE 4-3 Conversion d'un scénario de déploiement en une architecture de déploiement

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

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, équilibrateurs 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.

## 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 équilibrateurs 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 ES” à la page 53](#)).

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 ES, d'une mise à niveau de composants Java ES déjà installés ou d'un remplacement de composants tiers par Java ES. 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 lequel 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 55](#)).

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, etc.

## 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. Les outils du développeur Sun Java Studio simplifient 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 : 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

configuration 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.

## Postdéploiement

Au cours de la phase de **postdéploiement** du cycle de vie, vous exécutez une solution déployée dans un environnement de production. La phase d'opérations du cycle de vie comprend les tâches suivantes :

- **Contrôle** : comprend le contrôle régulier des performances et des fonctions du système.
- **Maintenance** : comprend les fonctions administratives quotidiennes, telles que l'ajout de nouveaux utilisateurs finals à un système, le changement de mots de passe, l'ajout de nouveaux utilisateurs administratifs, le changement des droits d'accès, la réalisation de sauvegardes régulières, etc.
- **Réglage des performances** : comprend 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** : comprend l'ajout de nouveaux composants Java ES à un système afin d'ajouter de nouvelles fonctionnalités ou de remplacer des composants non Java ES. 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.

## 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 ES.

### déploiement

Phase du cycle de vie d'une solution Java ES 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).

<b>deployment scenario</b> (scénario de déploiement)	<a href="#">logical architecture (architecture logique)</a> pour une solution Java ES 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 ou capacité latente. Un scénario de déploiement constitue le point de départ d'une conception de déploiement.
<b>développement</b>	Tâche du processus de déploiement d'une solution Java ES, au cours de laquelle les composants personnalisés d'une <a href="#">deployment architecture (architecture de déploiement)</a> sont programmés et testés.
<b>prédéploiement</b>	Phase du cycle de vie d'une solution Java ES au cours de laquelle les besoins d'exploitation sont convertis en un <a href="#">deployment scenario (scénario de déploiement)</a> : <a href="#">logical architecture (architecture logique)</a> et ensemble d'exigences de qualité de service auxquelles une solution doit satisfaire.
<b>postdéploiement</b>	Phase du cycle de vie d'une solution Java ES 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.
<b>reference deployment architecture</b> (architecture de déploiement de référence)	<a href="#">deployment architecture (architecture de déploiement)</a> 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.
<b>use case</b> (cas d'utilisation)	Tâche ou ensemble de tâches utilisateur spécifiques effectuées par une <a href="#">distributed enterprise application (application d'entreprise distribuée)</a> , et utilisées comme point de départ pour la conception, le test et la mesure des performances de l'application.

## Composants Java ES

---

Java ES est constitué d'un ensemble de [composants de produit](#) et de [composants partagés](#) interopérables assurant la prise en charge d'applications distribuées sur un réseau. Au cours de l'installation, le programme d'installation de Java ES présente des composants programmables, dont la plupart disposent de sous-composants programmables. Ces composants et sous-composants sont répertoriés dans cette annexe.

Cette annexe décrit brièvement les composants Java ES et sert de présentation générale. Pour obtenir des informations détaillées sur des composants spécifiques, reportez-vous à la documentation appropriée, disponible à l'adresse <http://docs.sun.com/app/docs/prod/entsys.5>. De nombreuses informations et ressources sur Java ES sont également disponibles à l'adresse <http://www.sun.com/bigadmin/hubs/javaes/>.

Les composants Java ES répertoriés dans cette annexe sont regroupés par catégorie et décrits dans les sections suivantes :

- “Composants de services système” à la page 71
- “Composants de qualité de service” à la page 75
- “Composants partagés” à la page 79

### Composants de services 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. Ces services, décrits dans la section “[À quoi sert Java Enterprise System ?](#)” à la page 17, comprennent les services de portail, les services d'identité et de sécurité, les services Web et d'application et les services de disponibilité. Ils sont décrits dans les sections suivantes :

- “Access Manager 7.1” à la page 72
- “Application Server Enterprise Edition 8.2” à la page 73
- “Directory Server Enterprise Edition 6.2” à la page 73

- “Java DB 10.2” à la page 74
- “Message Queue 3.7 UR2” à la page 74
- “Portal Server 7.1 Update 2” à la page 74
- “Service Registry 3.1 ” à la page 75
- “ Web Server 7.0 ” à la page 75

## Access Manager 7.1

Sun Java System Access Manager (Access Manager) intègre des services d'authentification et d'autorisation, des agents de stratégie ainsi qu'une fédération d'identité pour offrir une solution complète de protection des ressources réseau. Access Manager empêche l'accès non autorisé aux applications de services Web et au contenu Web, en fournissant une infrastructure permettant aux organisations de gérer 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 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.

Access Manager comprend les sous-composants suivants :

- **Services de base d'Access Manager** : 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.
- **Console d'administration d'Access Manager** : 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.
- **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 Distribution Authentication User Interface** : fournit une interface utilisateur activant un agent de stratégie ou une application déployé(e) dans une zone non sécurisée pour communiquer avec Access Manager Authentication Service installé dans une zone sécurisée du déploiement.
- **Access Manager Client SDK** : permet aux utilisateurs d'implémenter des applications autonomes pouvant accéder à un serveur Access Manager pour utiliser des services d'authentification, de connexion unique, d'autorisation, de contrôle, de connexion et le SAML (Security Assertion Markup Language).
- **Client de basculement de session Access Manager** : requis pour configurer le basculement de session Access Manager.

## Application Server Enterprise Edition 8.2

Sun Java System Application Server (Application Server) fournit une plate-forme compatible J2EE pour le développement et le déploiement d'applications Java côté serveur et de services Web. Parmi les fonctionnalités clés d'Application Server figurent celles de gestion des transactions évolutives, d'exécution de persistance gérée par conteneur, de performance des services Web, de clustering, d'état de session de haute disponibilité, de sécurité et d'intégration.

Application Server comprend les sous-composants suivants :

- **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.
- **Agent du nœud Application Server** : processus léger qui s'exécute sur toutes les machines hébergeant des instances de serveur et qui exécute plusieurs tâches administratives, notamment l'arrêt, le démarrage et le redémarrage d'instances de serveur.
- **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** : utilisé pour répartir de manière homogène la charge de travail entre plusieurs instances Application Server (autonomes ou clusterisées), augmentant ainsi la capacité de traitement total du système. Il est également utilisé pour activer des requêtes de basculement d'une instance de serveur à une autre.
- **Exemples d'applications** : installés lors de l'installation complète d'Application Server.

## Directory Server Enterprise Edition 6.2

Sun Java System Directory Server (Directory Server) offre un serveur d'annuaire LDAP fournissant 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 l'authentification des utilisateurs du réseau extranet.

Directory Server comprend les sous-composants suivants :

- **Directory Server 6.2 Core Server**. permet de stocker et de gérer des données d'identité de manière évolutive, sécurisée et souple.
- **Centre de contrôle du service d'annuaire** : fournit une interface d'administration basée sur navigateur permettant de configurer des services d'annuaire et des services proxy d'annuaire.
- **Utilitaire de ligne de commande Directory Server** : permet d'effectuer des tâches d'administration à partir de la ligne de commande.

- **Directory Proxy Server 6.2 Core Server.** optimise la sécurité en offrant des fonctionnalités d'annuaire virtuel et augmente la disponibilité et l'évolutivité du service d'annuaire.

## Java DB 10.2

Java DB fournit une base de données légère pour le développement d'applications Java. Java DB est la distribution prise en charge par Sun de la base de données de technologie open-source Apache Derby 100% Java. Java ES 5 fut la première version à inclure Java DB comme composant logiciel. Dans sa première version, Java DB était un composant partagé appelé Base de données Derby inclus dans Java ES 2005Q4.

Java DB comprend les sous-composants suivants :

- Java DB Client
- Java DB Server

## Message Queue 3.7 UR2

Sun Java System Message Queue (Message Queue) est une solution basée sur des normes permettant de résoudre le 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.

Le programme d'installation de Java ES fournit Message Queue en tant que composant installable unique.

## Portal Server 7.1 Update 2

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, le groupement, la sécurité, l'intégration et la recherche.

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

## Service Registry 3.1

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.

Service Registry comprend les sous-composants suivants :

- Prise en charge de Service Registry Client
- Prise en charge du déploiement de Service Registry

## Web Server 7.0

Sun Java System Web Server (Web Server) est un serveur Web sécurisé multitraitements et à unités d'exécution multiples créé sur des normes industrielles. Web Server offre performances, fiabilité, évolutivité et d'excellentes capacités de gestion pour les grandes et moyennes entreprises.

Web Server comprend les sous-composants suivants :

- Interface de ligne de commande Web Server
- Web Server Core
- Exemples Web Server

# Composants de qualité de service

Les composants de qualité de service Java ES améliorent la qualité des services fournis par les composants de services système ou 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.

Les composants prenant en charge les composants de service Java ES sont regroupés dans les catégories suivantes et décrits dans cette section :

- “Composants de disponibilité” à la page 76
- “Composants d'accès” à la page 78
- “Composants de contrôle” à la page 79

## 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 :

- “High Availability Session Store 4.4” à la page 76
- “Sun Cluster 3.1 8/06 et Agents Sun Cluster 3.1” à la page 76
- “Sun Cluster Geographic Edition 3.1 2006Q4” à la page 77

### High Availability Session Store 4.4

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.

### Sun Cluster 3.1 8/06 et Agents Sun Cluster 3.1

---

**Remarque** – Les composants Sun Cluster sont pris en charge sur la plate-forme Solaris uniquement.

---

Le logiciel Sun Cluster fournit des services haute disponibilité Évolutivité pour Java ES 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. Les agents Sun Cluster suivants sont inclus dans Java Enterprise System.

---

**Remarque** – Dans la liste suivante, *HA* signifie *haute disponibilité*.

---

- HA Application Server
- HA Message Queue
- HA Directory Server
- HA Messaging Server
- HA Application Server EE (HADB)
- HA/Scalable Web Server
- HA Instant Messaging
- HA Calendar Server
- HA Apache Tomcat
- HA Apache
- HA DHCP ;
- HA DNS
- HA MySQL ;
- HA Sun N1 Service Provisioning
- HA NFS
- HA Oracle
- HA Samba
- HA Sun N1 Grid Engine
- HA Solaris Containers

---

**Remarque** – La liste d'agents n'est pas la même sous SPARC et x86. Pour plus d'informations sur les agents Sun Cluster, reportez-vous à la documentation de Sun Cluster, disponible à l'adresse <http://docs.sun.com/app/docs/prod/entsys.5>.

---

## Sun Cluster Geographic Edition 3.1 2006Q4

Sun Cluster Geographic Edition est une extension en couches du logiciel Sun Cluster. Cette extension protège les applications contre les arrêts brutaux, à l'aide de plusieurs clusters répartis sur plusieurs emplacements géographiques et d'une infrastructure redondante répliquant les données entre ces clusters. Java ES 5 est la première version à inclure Sun Cluster Geographic Edition en tant que composant logiciel Java ES.

Sun Cluster Geographic Edition comprend les sous-composants suivants :

- Sun Cluster Geographic Edition Core
- Sun StorEdge Availability Suite
- Prise en charge de la réplication de données Hitachi Truecopy (SPARC uniquement)
- Réplication de données SRDF d'EMC

---

**Remarque** – Sun Cluster Geographic Edition n'est pas pris en charge sur Solaris x86.

---

## 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 :

- [“Portal Server Secure Remote Access 7.1” à la page 78](#)
- [“Web Proxy Server 4.0.5” à la page 78](#)

### Portal Server Secure Remote Access 7.1

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.

Portal Server Secure Remote Access comprend les sous-composants suivants :

- **Portal Server Secure Remote Access Core** : fournit les fonctionnalités de base.
- **Passerelle** : fournit l'interface et la barrière de sécurité entre les sessions utilisateur à distance en provenance d'Internet et un réseau intranet d'entreprise. La passerelle 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 et assure le contrôle des communications entre Portal Server et les diverses instances de passerelle.
- **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. Netlet permet l'accès et l'utilisation à distance de systèmes de fichiers et annuaires et assure la sécurité des communications entre l'applet Netlet et le navigateur client, la passerelle et les serveurs d'applications.
- **Proxy de réécriture** : permet d'assurer un trafic HTTP sécurisé entre la passerelle et les ordinateurs de l'intranet. Le 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.

### Web Proxy Server 4.0.5

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-feu

d'entreprise afin de réduire le nombre de requêtes vers les serveurs de contenu distants et à l'extérieur des pare-feu 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.

## Composants de contrôle

Sun Java System Monitoring Console 1.0 (Monitoring Console) est constituée d'un agent maître relié à tous les agents de nœud dans un déploiement Java ES. Monitoring Console est prise en charge par Sun Java System Monitoring Framework 2.0 (Monitoring Framework), un composant partagé fournissant l'instrumentation et l'agent de nœud nécessaires pour que chaque composant surveillé puisse exposer ses attributs en vue d'un contrôle. Chaque composant logiciel expose les objets représentant ses attributs observables et un agent de nœud regroupe la vue de plusieurs composants sur un hôte. Pour plus d'informations sur le contrôle, reportez-vous au *Sun Java Enterprise System 5 Update 1 Monitoring Guide*.

## 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. Ces composants sont des bibliothèques locales pouvant être partagées par tout composant de Java ES fonctionnant sur un ordinateur hôte donné. 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 ES comprend les composants partagés suivants :

- ACL (Apache Common Logging) 1.0.4
- ANT (outil de construction basé Jakarta ANT Java/XML) 1.6.5
- BDB (Base de données Berkeley) 4.2.52
- Common Agent Container 2.1 (Sun Cluster uniquement)
- Common Agent Container 2.1
- FastInfoSet 1.0.2
- ICU 3 (International Components for Unicode) 3.2
- J2SE (Java 2 Platform, Standard Edition) platform 5.0 Update 6
- JAF (JavaBeans™ Activation Framework) 1.0.3
- JATO (Java Studio Web Application Framework) 2.1.5
- JavaHelp™ 2.0

- JavaMail™ API 1.3.2
- JAXB (Java Architecture for XML Binding - Architecture Java pour liaison XML)
- JAXP (Java API for XML Processing - API Java pour traitement XML) 1.3.1\_01
- JAXR (Java API for XML Registries) 1.0.8
- JAXRPC (Java API for XML-based Remote Procedure Call) 1.1.3\_01
- JAXWS (Java API for Web Services) 2.0
- JDMK (Java Dynamic Management Kit) 5.1\_03
- JSS (Java Security Services) 4.2.4
- JSS3 (Network Security Services for Java) 3.1.11
- JSTL (JavaServer Pages™ Standard Tag Library) 1.0.6
- KTSE (KT Search Engine) 1.3.4
- LDAP C SDK 6.0
- LDAP Java SDK 4.19
- MA Core (Mobile Access Core) 6.3.1
- NSPR (Netscape Portable Runtime) 4.6.3
- NSPRD (Netscape Portable Runtime Development) 4.6
- NSS (Network Security Services) 3.11
- NSSU (Network Security Service Utilities) 3.11
- SAAJ (SOAP with Attachments API for Java) 1.3
- SASL (Simple Authentication and Security Layer) 2.19
- Sun Explorer Data Collector (SE Solaris seulement) 4.3.1
- Sun Java System Monitoring Framework 2.0 (prend en charge Monitoring Console 1.0)
- Sun Java Web Console 3.0.2
- WSCL (Web Services Common Library) 2.0
- XWSS (XML Web Services Security) 2.0

# Index

---

## A

### Access Manager

- Composant de services système, 21
- Description, 72
- Service d'infrastructure, 38

### Améliorations, *Voir* Scénarios d'adoption

### Annuaire

- Définition, 59
- sous la forme de magasins de données utilisateur, 56

### Apache Derby, 21

### Application Platform Suite, 24

### Application Server

- Composant de services système, 21
- Description, 73
- Service d'infrastructure, 38

### Applications

#### Distribuées

*Voir* Applications d'entreprise distribuées

#### Entreprise

*Voir* Applications d'entreprise distribuées

### Applications d'entreprise distribuées

- À propos de, 17
- Définition, 31
- Infrastructure, 18

### Architecture

- Définition, 50
- Déploiement, 64-66
- Dimensions
  - Voir* Dimensions architecturales
- Présentation, 33
- Solution, 33

### Architecture de déploiement, Créée par, 64-66

### Architectures de déploiement

- Définition, 51
- présentation, 33
- Relation avec les architectures à plusieurs niveaux, 42

### Architectures de déploiement de référence,

- Définition, 70

### Architectures logiques

- Définition, 51
- Exemple, 48-49
- Niveaux de service de l'infrastructure, 35
- Présentation, 33

### authentification, 57

### Autorisation, 58-59

### Availability Suite, 24

## B

### Base de données Derby, 21

## C

### Calendar Server, 25

### Cas d'utilisation

- Définition, 70
- Présentation, 63

### Catégories d'utilisateur

- Administrateur délégué, 29
- Administrateur système, 29
- Administrateur système spécialisé, 29
- Analyste système, 28

- Catégories d'utilisateur (*Suite*)
    - Architecte, 28
    - Ingénieur de projet, 29
    - Intégrateur système, 29
    - Planificateur, 28
    - Responsable informatique, 29
  - Clients
    - Composants de services système, 20
    - Définition, 50
  - Clusters
    - Voir* Sun Cluster
  - Communications Express, 25
  - Communications Suite, 19, 25
  - Composants
    - Accès, 23
    - Communications, 25
    - Contrôle, 23
    - Définition, 31
    - Dépendances, 39-40
    - Détection des versions installées, 54
    - Disponibilité, 22-23
    - Distribués, 17
    - EJB, 41
    - J2EE, 41
    - JSP, 41
    - Partagés, 23, 79-80
    - Produit, 32
    - Qualité de service, 22-23, 75-79
    - Service du système, 71-75
    - Service système, 20-22
    - Services d'infrastructure, 38
    - Servlet, 41
    - Système
      - Voir* Composants système
  - Composants d'accès
    - Description, 78-79
    - Présentation, 23
  - Composants d'application
    - Dans les architectures de niveaux logiques, 40
    - Définition, 50
  - Composants de communication, 19, 25, 43
  - Composants de disponibilité
    - Description, 76-78
    - Présentation, 22-23
  - Composants de produit, Définition, 32
  - Composants de qualité de service
    - Définition, 32
    - Description, 75-79
    - Présentation, 22-23
  - Composants de service système
    - Dépendances, 39-40
    - Présentation, 20-22
  - Composants de services système, Définition, 32
  - Composants du système
    - Composants partagés, 79-80
    - Service du système, 71-75
  - Composants EJB, 41
  - Composants Java servlet, 41
  - Composants JSP, 41
  - Composants partagés, 79
    - Définition, 32
    - Présentation, 23
  - Composants système
    - À propos de, 20
    - Composants de qualité de service, 22-23
    - Composants de service système
      - Voir* Composants de service du système
    - Composants partagés, 23
    - Définition, 32
  - Connexion unique
    - Définition, 60
    - Fonction Java ES, 21, 57
    - Implémentation, 58
    - Niveaux de service d'infrastructure, 37
  - Console de contrôle, 23
    - Description, 79
    - Suites, 24-26
  - Contrôle, à propos de, 23, 79
  - Contrôle de dépendance, programme d'installation, 54
  - Cycle de vie, phases
    - Déploiement, 28, 64
    - Postdéploiement, 28, 69
    - Prédéploiement, 63
- D**
- Delegated Administrator, 25
  - Dépendances, 39-40, 54

- Déploiement
    - Architecture, 64
    - Conception, 64-66
    - Définition, 69
    - Développement et personnalisation, 67
    - Implémentation, 66-69
    - Mise en production, 68-69
    - Phase de cycle de vie, 64-69
    - Phases du cycle de vie, 64
    - Scénarios
      - Voir* Scénarios de déploiement
    - Test de prototype, 68
  - Détection des logiciels installés, 54
  - Développement
    - Définition, 70
    - Tâche de déploiement, 67
  - Dimensions architecturales
    - Dépendances des services d'infrastructure, 35
    - Niveaux logiques, 40
    - Qualité de service, 43-47
    - Synthèse, 47
  - Directory Server
    - Composant de services système, 21
    - Description, 73
    - Service d'infrastructure, 38
  - Disponibilité
    - Exigences, 44
    - Services, 19, 46, 76
  - Disponibilité, Exigences, 45-46
  - Distribuées
    - Applications
      - Voir* Applications d'entreprise distribuées
  - Distribués
    - Services
      - Voir* Services distribués
- E**
- Entrée utilisateur, 56
  - Évolutivité
    - Exigences, 44, 45-46
    - Services, 46, 76
  - Exigences d'entretien, 44, 45-46
  - Exigences de capacité latente, 44
  - Exigences de performance, 44, 45-46
  - Exigences de qualité de service
    - Capacité latente, 44
    - Disponibilité, 44
    - Entretien, 45-46
    - Évolutivité, 44
    - Performances, 44
    - Sécurité, 44
- F**
- Fonctionnalité d'intégration, Programme d'installation intégré, 53-55
  - Fonctionnalités d'intégration, Identité et sécurité, 55-59
  - Fonctions d'intégration
    - Composants partagés, 19
    - Identité et sécurité, 19
    - Programme d'installation intégré, 19
  - Formation, scénarios d'adoption de Java ES, 30
- G**
- Glossaire, lien, 13
- H**
- High Availability Session Store
    - Composant de qualité de service, 22
    - Description, 76
- I**
- Identité
    - Gestion, 55
    - Services, 19, 55-59
    - Utilisateur unique, 56-57
  - Identité unique
    - Définition, 60
    - Présentation, 56
  - Identity Management Suite, 25

**Infrastructure**

- Applications d'entreprise distribuées, 18

- Dépendances des services

  - Voir* Services distribués

- Instant Messaging, 25

**Intégration**

- Scénarios d'adoption de Java ES, 30

- Services, 37

**J****J2EE**

- Composants, 41

- Modèle de composant distribué, 41

- Plate-forme, 22

**Java DB**

- Composant de services système, 21

- Description, 74

- Java ES documentation, 12

- JMS (Java Message Service), 21

- JSS (Java Security Services), 23

**L**

- LDAP, 41, 59

**M**

- Matériel, scénarios d'adoption de Java ES, 30

**Message Queue**

- Composant de services système, 21

- Description, 74

- Service d'infrastructure, 38

- Messaging Server, 25

- Migration, scénarios d'adoption de Java ES, 30

- Mise en production, 68-69

**N****Niveau logique**

- Présentation, 40

**Niveau logique (*Suite*)**

- Service d'entreprise, 41

**Niveaux, logiques**

- Architecture d'application, 40

- Client, 40

- Données, 41

- NSPR (Netscape Portable Runtime), 23

- NSS (Network Security Services), 23

**P**

- Phases du cycle de vie, Prédéploiement, 28

- Plate-forme J2ME, 40

**Portal Server**

- Composant de services système, 21

- Description, 74

- Service d'infrastructure, 38

**Portal Server Secure Remote Access**

- Composant de qualité de service, 23

- Composant système, 45

- Description, 78

**Postdéploiement**

- Définition, 70

- Phases du cycle de vie, 69

**Prédéploiement**

- Définition, 70

- Phases du cycle de vie, 63

**Produits Sun Java System****Access Manager**

- Voir* Access Manager

**Application Server**

- Voir* Application Server

**Directory Server**

- Voir* Directory Server

**High Availability Session Store**

- Voir* High Availability Session Store

**Java DB**

- Voir* Java DB

**Message Queue**

- Voir* Message Queue

**Portal Server**

- Voir* Portal Server

**Portal Server, Secure Remote Access**

- Voir* Portal Server, Secure Remote Access

Produits Sun Java System (*Suite*)

- Service Registry
  - Voir* Service Registry
- Sun Cluster
  - Voir* Sun Cluster
- Sun Cluster Geographic Edition
  - Voir* Sun Cluster Geographic Edition
- Web Proxy Server
  - Voir* Web Proxy Server
- Web Server
  - Voir* Web Server

Profils utilisateur, 28

Programme de désinstallation, 55

Prototype, 68

Provisioning d'utilisateurs, 66

**Q**

## Qualité des exigences de service

- Disponibilité, 45-46
- Entretien, 44
- Évolutivité, 45-46
- Performance, 45-46
- Sécurité, 45-46

**S**

## Scénarios d'adoption, Java ES

- Amélioration, 29
- À propos de, 29-31
- Définition, 31
- Extension, 29
- Mise à niveau, 30
- Nouveau système, 29

## Scénarios de déploiement

- Définition, 70
- Présentation, 63

## Sécurité

- Exigences, 44, 45-46
- Services, 19
- Services de stratégie, 37

## Serveurs

- Autonomes, 41

Serveurs (*Suite*)

- Définition, 51
- Service, Web, 41
- Service Registry
  - Composant de services système, 21
  - Description, 75
- Services
  - Contrôle, 19
  - Définition, 32
  - Évolutivité, 46, 76
  - Haute disponibilité, 46, 76
  - Infrastructure, 18
    - Voir* Services d'infrastructure distribués

Services d'accès, 19

Services d'application, 19, 35

Services d'entreprise

Définition, 50

Niveau présentation, 41

Services d'exécution, 37

Services de collaboration, 18

Services de collaboration utilisateur, 37

Services de communication, 18

Services de messagerie, 36

Services de persistance, 36

Services de plate-forme, 35

Services de portail, 18

Services de transport réseau, 36

Services distribués

Accès, 19

Communication et collaboration, 18

Contrôle, 19

Disponibilité, 19

Exécution, 19

Identité, 19

Infrastructure, 18

Intégration, 37

Intermédiaires, 35

Messagerie, 36

Niveau application, 35

Persistance, 36

Plate-forme, 35, 36

Portail, 18

Présentation, 18

Sécurité, 19, 37

Services distribués (*Suite*)

Transport réseau, 36

Web, 19

Services du système d'exploitation, 36

Services intermédiaires, 35

Services système

À propos, 20

Définition, 32

Services Web, 19

Composants J2EE, 41

Définition, 51

Solutions, Java ES

Architecture, 33

Cycle de vie, 26-29

Exemple, 48

Personnalisées et standard, 47

Spécifications d'implémentation, 66

Stratégies

Autorisation, 58

Définition, 59

Suites, 24

Sun Cluster

Agents, 46

Composant de qualité de service, 22

Description, 76

Service de disponibilité, 46-47

Sun Cluster Geographic Edition

Composant de qualité de service, 22

Description, 77

Système

Composants

*Voir* Composants système

Configuration, 54-55

Services, 17-19

**W**

Web Infrastructure Suite, 26

Web Proxy Server, Composant de qualité de service, 23

Web Server

Composant de services système, 22

Description, 75

Service d'infrastructure, 38

**T**

Tâches, Java ES, 26, 61

**U**

Utilisateurs, Applications distribuées, 17

Utilisateurs finals, Définition, 31