



Guide de planification du déploiement de Sun Java Enterprise System 2005Q4

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

Référence : 819-3453
Octobre 2005

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

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

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

Sun, Sun Microsystems, le Sun logo, docs.sun.com, AnswerBook, AnswerBook2, Java, et Solaris sont des marques de fabrique ou déposées de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

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

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

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



051220@13215



Table des matières

Préface	11
1 Introduction à la planification du déploiement	17
À propos de Java Enterprise System	17
Services système	17
Services intégrés et services personnalisés	19
Migration vers Java Enterprise System	19
À propos de la planification du déploiement	20
Cycle de vie de la solution	20
Phase d'analyse d'exploitation	22
Phase de spécification technique	22
Phase de conception logique	23
Phase de conception du déploiement	23
Phase d'implémentation	24
Phase de fonctionnement	24
2 Analyse d'exploitation	27
À propos de l'analyse d'exploitation	27
Définition des exigences de l'entreprise	28
Définition des objectifs d'exploitation	28
Identification des besoins des utilisateurs	29
Connaissance de la culture de l'entreprise	30
Approche incrémentielle	32
Contrats de niveau de service (SLA)	32
Définition des contraintes d'exploitation	33
Problèmes de migration	33

	Établissement de la planification	33
	Limitations budgétaires	33
	Coût de possession	34
3	Exigences techniques	35
	À propos des exigences techniques	35
	Analyse d'utilisation	36
	Cas d'utilisation	38
	Exigences de qualité de service	39
	Performances	40
	Disponibilité	40
	Évolutivité	43
	Exigences liées à la sécurité	44
	Capacité latente	46
	Exigences d'entretien	46
	Exigences de niveau de service	47
4	Conception logique	49
	À propos des architectures logiques	49
	Conception d'une architecture logique	51
	Composants Java Enterprise System	51
	Dépendances entre composants	52
	Prise en charge d'un conteneur Web	55
	Services logiques distincts fournis par Messaging Server	55
	Composants d'accès	56
	Conception d'architecture à plusieurs niveaux	56
	Exemple d'architecture logique	58
	Exemple de Messaging Server	58
	Exemple de communications basées sur les identités	63
	Zones d'accès	67
	Scénario de déploiement	68
5	Conception du déploiement	69
	À propos de la conception du déploiement	70
	Approbation du projet	70
	Résultats de la conception du déploiement	70
	Facteurs influant sur la conception du déploiement	71

Méthodologie de conception du déploiement	73
Estimation de la puissance de traitement	74
Exemple d'estimation de la puissance de traitement	75
Estimation de la puissance de traitement requise pour les transactions sécurisées	80
Estimation du nombre de CPU pour les transactions sécurisées	80
Matériel dédié au traitement des transactions SSL	82
Identification des stratégies de disponibilité	83
Stratégies de disponibilité	83
Exemples de conceptions de disponibilité	87
Identification des stratégies d'évolutivité	91
Capacité latente	92
Exemple d'évolutivité	92
Identification des goulots d'étranglement des performances	93
Optimisation de l'accès aux disques	95
Conception pour une utilisation optimale des ressources	96
Gestion des risques	97
Exemple d'architecture de déploiement	98
6 Implémentation d'une conception de déploiement	101
À propos de l'implémentation des conceptions de déploiement	101
Installation et configuration du logiciel	102
Développement de pilotes et de prototypes	103
Test des pilotes et des prototypes	103
Passage à un déploiement de production	104
Index	107

Liste des tableaux

TABEAU 1-1	Catégories de services Java Enterprise System	18
TABEAU 3-1	Facteurs à prendre en compte lors de l'analyse d'utilisation	36
TABEAU 3-2	Qualités système ayant un impact sur les exigences de qualité de service	39
TABEAU 3-3	Interruption d'activité non planifiée d'un système fonctionnant toute l'année (8 760 heures)	41
TABEAU 3-4	Disponibilité des services par ordre de priorité	42
TABEAU 3-5	Facteurs d'évolutivité	44
TABEAU 3-6	Facteurs concernant les exigences d'entretien	46
TABEAU 4-1	Dépendances entre composants de Java Enterprise System	53
TABEAU 4-2	Configurations de Messaging Server	55
TABEAU 4-3	Composants de Java Enterprise System fournissant un accès distant	56
TABEAU 4-4	Niveaux logiques dans une architecture à plusieurs niveaux	57
TABEAU 4-5	Composants de l'architecture logique de Messaging Server	59
TABEAU 4-6	Zones d'accès sécurisé et leurs composants	68
TABEAU 5-1	Estimation du nombre de CPU pour les composants contenant des points d'entrée utilisateur	76
TABEAU 5-2	Estimation du nombre de CPU pour les composants de support	77
TABEAU 5-3	Ajustement de l'estimation du nombre de CPU pour les charges de pointe	78
TABEAU 5-4	Ajustement de l'estimation du nombre de CPU pour les composants de support	79
TABEAU 5-5	Modification du nombre de CPU estimé pour les transactions sécurisées	81
TABEAU 5-6	Ajustement de l'estimation du nombre de CPU pour les composants de support	87
TABEAU 5-7	Points d'accès aux données	94

TABLEAU 5-8 Gestion des ressources 96

Liste des figures

FIGURE 1-1	Cycle de vie de la solution	21
FIGURE 4-1	Composants Java Enterprise System	52
FIGURE 4-2	Dépendances entre composants de Java Enterprise System	54
FIGURE 4-3	Modèle d'architecture à plusieurs niveaux	57
FIGURE 4-4	Architecture logique d'un déploiement Messaging Server	59
FIGURE 4-5	Composants logiques placés dans des zones d'accès	67
FIGURE 5-1	Architecture logique d'un scénario de communications basées sur les identités	76
FIGURE 5-2	Système à serveur unique	84
FIGURE 5-3	Système à basculement N+1 avec deux serveurs	85
FIGURE 5-4	Équilibrage de charge et basculement entre deux serveurs	85
FIGURE 5-5	Répartition de la charge entre n serveurs	86
FIGURE 5-6	Conception avec basculement utilisant le logiciel Sun Cluster	89
FIGURE 5-7	Exemple de réplication monomaître	90
FIGURE 5-8	Exemple de réplication multimaître	91
FIGURE 5-9	Exemples de mise à l'échelle horizontale et verticale	93
FIGURE 5-10	Exemple d'architecture de déploiement	99

Préface

Le manuel *Guide de planification du déploiement de Sun Java Enterprise System 2005Q4* indique brièvement les modalités de planification et de conception des solutions de déploiement d'entreprise basées sur Sun Java™ Enterprise System. Il présente quelques principes et concepts fondamentaux de la planification et de la conception de déploiement, traite du cycle de vie d'une solution, et, par conséquent, des phases et tâches d'un projet de conception de déploiement, et fournit des exemples et des stratégies de haut niveau à utiliser lors de la planification de solutions de déploiement à l'échelle de l'entreprise à l'aide de Java Enterprise System (Java ES).

Public concerné par ce manuel

Il est principalement destiné au personnel responsable du déploiement et aux planificateurs d'entreprise chargés de l'analyse et de la conception de déploiements d'entreprise. Il s'adresse également aux intégrateurs système et aux autres personnes responsables de la conception et de l'implémentation de divers aspects d'une application d'entreprise.

Préalable à la lecture de ce manuel

Ce guide part du principe que vous êtes déjà familiarisé avec la conception et l'installation d'applications à l'échelle de l'entreprise et que vous avez pris connaissance du document *Présentation technique de Sun Java Enterprise System 2005Q4*.

Organisation du manuel

Ce manuel s'appuie sur le cycle de vie d'une solution pour décrire les différentes phases de la planification du déploiement. Le [Chapitre 1](#) décrit le cycle de vie de la solution.

Documentation relative à Java ES

La documentation Java ES concerne la planification du déploiement et l'installation du système. L'URL correspondant à la documentation système est le suivant : <http://docs.sun.com/coll/1286.1>. Pour une introduction à Java ES, reportez-vous aux manuels en respectant l'ordre dans lequel ils apparaissent dans le tableau suivant :

TABLEAU P-1 Documentation Java Enterprise System

Titre du document	Sommaire
<i>Notes de version de Sun Java Enterprise System 2005Q4</i>	Ce manuel contient les informations les plus récentes sur Java ES, y compris les problèmes connus. De plus, chaque composant dispose de notes de version qui lui sont propres.
<i>Présentation de la documentation de Sun Java Enterprise System 2005Q4</i>	Ce manuel décrit toute la documentation relative à Java ES et à ses composants.
<i>Présentation technique de Sun Java Enterprise System 2005Q4</i>	Ce manuel présente les principes techniques et conceptuels de Java ES. Il décrit les composants, l'architecture, les processus et les fonctions.
<i>Guide de planification du déploiement de Sun Java Enterprise System 2005Q4</i>	Ce manuel indique brièvement les modalités de planification et de conception des solutions de déploiement d'entreprise basées sur Java ES. Il présente quelques principes et concepts fondamentaux de la planification et de la conception de déploiement, traite du cycle de vie d'une solution et fournit des exemples et des stratégies de haut niveau à utiliser lors de la planification de solutions basées sur Java ES.

TABLEAU P-1 Documentation Java Enterprise System (Suite)

Titre du document	Sommaire
<i>Guide de planification de l'installation de Java ES System 2005Q4</i>	Ce manuel vous aide à développer les spécifications d'implémentation pour le matériel, le système d'exploitation et les différents aspects réseau de votre déploiement Java ES. Il décrit les aspects à prendre en compte dans votre plan d'installation et de configuration tels que les dépendances entre les composants.
<i>Guide d'installation de Sun Java Enterprise System 2005Q4 pour UNIX</i>	Ce manuel vous guide tout au long de la procédure d'installation de Java ES pour le système d'exploitation Solaris ou Linux. Il explique également comment configurer les composants après l'installation et vérifier leur bon fonctionnement.
<i>Référence de l'installation de Java ES System 2005Q4</i>	Ce manuel fournit des informations complémentaires concernant les paramètres de configuration et inclut des feuilles de travail à utiliser lors de l'établissement de votre plan de configuration. Il répertorie les données de références telles que les répertoires par défaut et les numéros de ports.
<i>Sun Java Enterprise System 2005Q1 Deployment Example Series: Evaluation Scenario</i>	Ce manuel décrit les procédures d'installation de Java ES sur un ordinateur, de mise en place d'un ensemble de services fondamentaux, réseau et partagés, et de configuration des comptes utilisateur pouvant accéder à ces services.
<i>Guide de mise [00e0] niveau de Sun Java Enterprise System 2005Q4</i>	Ce manuel fournit les instructions nécessaires à la mise à niveau de Java ES pour le système d'exploitation Solaris ou Linux.
<i>Sun Java Enterprise System Glossary</i>	Il contient les définitions des termes utilisés dans la documentation Java ES.

Conventions typographiques

Les conventions typographiques employées dans le présent guide sont décrites dans le tableau ci-dessous.

TABLEAU P-2 Conventions typographiques

Type de caractères	Signification	Exemple
AaBbCc123	Noms de commande, noms de fichier, chemins d'accès aux répertoires, éléments affichés sur l'écran de l'ordinateur	Modifiez le fichier <code>.login</code> . Pour établir la liste de tous les fichiers, entrez <code>ls -a</code> . <code>machine_name%</code> Vous avez du courrier.
AaBbCc123	Ce que vous entrez, et non ce qui apparaît sur l'écran	<code>machine_name%</code> su Mot de passe :
<i>AaBbCc123</i>	Marque de réservation à remplacer par un nom ou une valeur réels	La commande permettant de supprimer un fichier est la suivante : <code>rm filename</code> .
<i>AaBbCc123</i>	Titres d'ouvrage, mots ou termes nouveaux importants (Remarque : en ligne, certains de ces éléments s'affichent en gras)	Lisez le chapitre 6 du <i>Guide de l'utilisateur</i> . Un <i>cache</i> est une copie stockée en local. Ne <i>pas</i> sauvegarder ce fichier.

Invites de Shell dans des exemples de commande

Le tableau suivant présente les invites système par défaut ainsi que les invites du superutilisateur.

TABLEAU P-3 Invites de Shell

Shell	Invite
C shell sur les systèmes UNIX et Linux	<code>machine_name%</code>
Superutilisateur C shell sur les systèmes UNIX et Linux	<code>machine_name#</code>
Bourne shell et Korn shell sur les systèmes UNIX et Linux	<code>\$</code>
Superutilisateur Bourne shell et Korn shell sur les systèmes UNIX et Linux	<code>#</code>
Ligne de commande Microsoft Windows	<code>C:\</code>

Conventions de symboles

Le tableau suivant présente les symboles susceptibles d'être utilisés dans ce manuel.

TABLEAU P-4 Conventions de symboles

Symbole	Description	Exemple	Signification
[]	Contient des arguments optionnels et des options de commande.	ls [-l]	L'option -l n'est pas obligatoire.
{ }	Délimite un ensemble de choix pour une option de commande obligatoire.	-d {y n}	L'option -d nécessite l'utilisation de l'argument y ou de l'argument n.
\${ }	Indique une référence de variable.	\${com.sun.javaRoot}	Fait référence à la valeur de la variable com.sun.javaRoot.
-	Joint plusieurs touches utilisées simultanément.	Ctrl-A	Appuyez sur la touche Ctrl tout en maintenant la touche A enfoncée.
+	Joint plusieurs touches utilisées de manière consécutive.	Ctrl+A+N	Appuyez sur la touche Ctrl, relâchez-la, puis appuyez sur les touches suivantes.
→	Indique une sélection d'options de menu dans une interface graphique.	Fichier → Nouveau → Modèles	Dans le menu Fichier, sélectionnez Nouveau. Dans le sous-menu Nouveau, sélectionnez Modèles.

Accès aux ressources Sun en ligne

Le site [Web docs.sun.com](http://www.java.sun.com)SM vous permet d'accéder à la documentation technique Sun en ligne. Vous pouvez parcourir les documents archivés ou rechercher un titre de document ou un sujet spécifique. Les manuels sont disponibles en ligne aux formats PDF et HTML. Ces deux formats sont compatibles avec les aides technologiques proposées aux utilisateurs souffrant d'un handicap physique.

Pour accéder aux ressources Sun suivantes, allez à la page <http://www.sun.com> :

- Téléchargement
- Services et solutions
- Support client (incluant les patchs et les mises à jour)

- Formation
- À propos de Sun
- Communautés (ex. : le portail Développeurs de Sun)

Références aux sites Web tiers

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

Remarque – Sun décline toute responsabilité quant à la disponibilité des sites Web tiers mentionnés. Sun ne garantit pas le contenu, la publicité, les produits et autres matériaux disponibles sur ces sites ou dans ces ressources, ou accessibles par leur intermédiaire, et ne saurait en être tenu 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 partager vos commentaires, accédez à la page <http://docs.sun.com> et cliquez sur Envoyer des commentaires. Dans le formulaire en ligne, indiquez le titre complet et le numéro de référence du document. Le numéro de référence est constitué de 7 ou 9 chiffres et se trouve sur la page de titre du manuel ou dans l'URL du document. Par exemple, le numéro de référence de ce manuel est 819-3453.

Introduction à la planification du déploiement

Ce chapitre décrit brièvement Sun Java™ Enterprise System (Java ES), aborde les concepts de planification du déploiement et présente le cycle de vie de la solution où sont énumérées les différentes étapes de planification et de conception des systèmes logiciels d'entreprise. Ce chapitre se compose des sections suivantes :

- “À propos de Java Enterprise System” à la page 17
- “À propos de la planification du déploiement” à la page 20

À propos de Java Enterprise System

Java Enterprise System est une infrastructure logicielle qui fournit les logiciels intermédiaires nécessaires à la prise en charge des applications d'entreprise distribuées dans un environnement réseau ou Internet. Les composants de Java Enterprise System fournissant les services sont installés par l'intermédiaire d'un programme d'installation commun et synchronisés par rapport à un ensemble de bibliothèques partagées. Ils partagent également un système intégré de gestion des identités utilisateur et de gestion de la sécurité.

Services système

Les principaux services d'infrastructure pris en charge par les composants de Java Enterprise System peuvent être classés selon les catégories suivantes :

- **Services de portail** : ils permettent aux employés amenés à se déplacer, aux télétravailleurs, aux responsables de la base de connaissances, aux partenaires commerciaux, aux fournisseurs et aux clients d'accéder de façon sécurisée, via Internet, au portail personnalisé de leur entreprise, depuis n'importe quel emplacement situé en dehors du réseau de l'entreprise. Ces services garantissent

aux communautés d'utilisateurs un accès au portail de l'entreprise à tout moment et en tout lieu, en fournissant des fonctions d'intégration, d'agrégation, de personnalisation, de sécurité, d'accès mobile et de recherche.

- **Services de communication et de collaboration** : ces services permettent un échange d'informations sécurisé entre les diverses communautés d'utilisateurs. La messagerie, la collaboration en temps réel ainsi que la planification de calendriers sont des exemples de capacités particulières applicables à l'environnement de travail de l'utilisateur.
- **Services réseau d'identité et de sécurité** : ces services améliorent la sécurité et la protection des ressources d'informations clés de l'entreprise en garantissant l'application de stratégies de contrôle d'accès appropriées pour la totalité des communautés, des applications et des services. 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 Web et applicatifs** : ces services permettent aux composants distribués de communiquer entre eux et de prendre en charge le développement, le déploiement et la gestion des applications pour une large palette de serveurs, de clients et de périphériques. Ils sont basés sur la plate-forme Java 2, Enterprise Edition (J2EE™).
- **Services de disponibilité** : ces services fournissent une disponibilité et une évolutivité presque permanentes des applications et des services Web.

Le tableau ci-dessous récapitule les catégories de services décrites précédemment et précise les composants de Java Enterprise System fournissant ces services pour chaque catégorie.

TABLEAU 1-1 Catégories de services Java Enterprise System

Catégorie de service	Composants Java Enterprise System
Services de portail	Portal Server, Portal Server Secure Remote Access, Access Manager, Directory Server, Application Server ou Web Server
Services de communication et de collaboration	Messaging Server, Calendar Server, Instant Messaging, Access Manager, Directory Server, Application Server ou Web Server
Services d'identité réseau	Access Manager, Directory Server, Web Server
Services Web et applicatifs	Application Server, Message Queue, Web Server
Services de disponibilité	Sun Cluster, Sun Cluster Agents

Pour plus d'informations sur les services, les composants et les concepts architecturaux de Java Enterprise System, reportez-vous au manuel *Présentation technique de Sun Java Enterprise System 2005Q4*.

Services intégrés et services personnalisés

Les solutions de déploiement fondées sur Java Enterprise System peuvent se classer selon deux grandes catégories :

- **Déploiement de type 80-20** : ces solutions font principalement appel à des services fournis par Java Enterprise System dans une proportion d'au moins 80 %.
- **Déploiement de type 20-80** : ces solutions font appel à de nombreux services personnalisés et applications tierces.

Les types de déploiement 80-20 et 20-80 sont des catégories générales. Le pourcentage exact du type de services offerts importe peu. Il permet néanmoins d'indiquer le taux de personnalisation d'une solution.

Le système Java Enterprise System est particulièrement adapté aux déploiements 80-20 en raison du large éventail de services qu'il propose. Par exemple, il est relativement simple de déployer un système de communications ou de portail à l'échelle de l'entreprise à l'aide des services fournis par Java Enterprise System.

Dans le cas des déploiements nécessitant des services et des applications personnalisés, Java Enterprise System prend en charge la création et l'intégration de ces éléments.

La plupart des catégories de services mentionnées à la section "[Services système](#)" à la [page 17](#) peuvent être utilisées dans le cadre d'un déploiement de type 80-20. Par exemple, les services de communication et de collaboration fournissent les services de messagerie, de calendrier et de messagerie instantanée aux utilisateurs finals et leur permettent de regrouper et de personnaliser les données. De même, les services d'identité réseau et de portail d'entreprise permettent d'installer et de configurer des applications à l'échelle de l'entreprise sans avoir à développer ni à intégrer des services personnalisés.

Les solutions d'entreprise exigeant des services de plate-forme J2EE adaptés peuvent tirer partie d'Application Server, de Message Queue ou de Web Server fournis par les services Web et applicatifs de Java Enterprise System.

Le pourcentage de services personnalisés requis peut varier considérablement d'un déploiement d'entreprise à l'autre. L'interopérabilité entre les services de Java Enterprise System vous permet de créer votre propre gamme de services adaptée aux besoins de votre entreprise.

Migration vers Java Enterprise System

La planification, la conception et l'implémentation d'une solution d'entreprise reposant sur Java Enterprise System dépendent beaucoup de votre stratégie de déploiement en cours. La planification, la conception et l'implémentation dans une entreprise envisageant une solution de déploiement pour la première fois sont

déterminées en grande partie par les besoins propres à celle-ci. Toutefois, les solutions de déploiement implémentées pour la première fois ne sont pas le cas de figure le plus répandu. Il est plus courant de rencontrer le cas de figure où Java Enterprise System est utilisé pour consolider les solutions d'entreprise existantes ou pour effectuer une mise à niveau des composants de Java Enterprise System.

Le remplacement ou la mise à niveau de solutions existantes exige des étapes supplémentaires de planification, de conception et d'implémentation pour assurer la préservation des données existantes et la mise à niveau correcte des logiciels vers les versions en cours. À mesure que vous suivez les étapes d'analyse et de conception décrites dans ce guide, gardez à l'esprit que vous devez préparer et planifier le remplacement et la mise à niveau des logiciels existants.

Pour plus d'informations sur la mise à niveau vers la version la plus récente de Java Enterprise System et sur les stratégies de migration à partir d'autres applications, consultez le manuel *Java Enterprise System Upgrade and Migration Guide*.

À propos de la planification du déploiement

La planification du déploiement est une étape essentielle de l'implémentation d'une solution Java Enterprise System. Chaque entreprise définit ses propres objectifs, exigences et priorités. Une planification réussie commence par l'analyse des objectifs de l'entreprise et des exigences requises pour atteindre ces objectifs. Les exigences de l'entreprise doivent ensuite être traduites sous forme d'exigences techniques qui serviront de base à la conception et à l'implémentation d'un système capable de répondre aux attentes de l'entreprise.

Pour être couronnée de succès, une planification de déploiement doit être préparée, analysée et conçue avec soin. Toute erreur ou maladresse commise au cours de la planification peut faire échouer le projet. Un système n'ayant pas fait l'objet d'une planification rigoureuse peut connaître de graves problèmes par la suite. Il peut par exemple offrir des performances insuffisantes et être difficile à entretenir. Son fonctionnement est susceptible de revenir cher, il peut entraîner une dispersion des ressources ou être incapable d'évoluer pour faire face à l'augmentation des besoins.

Cycle de vie de la solution

Le cycle de vie de la solution, illustré par le schéma ci-dessous, correspond aux étapes de planification, de conception et d'implémentation d'une solution logicielle d'entreprise reposant sur Java Enterprise System. Utilisez-le comme fil directeur pour votre projet de déploiement.

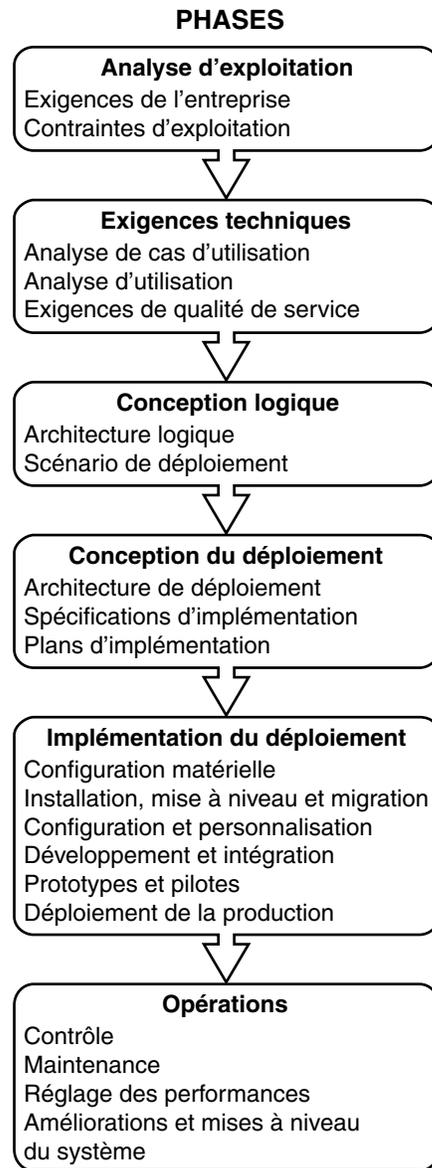


FIGURE 1-1 Cycle de vie de la solution

Le cycle de vie est constitué de phases devant respecter un ordre précis. Une phase se compose d'une combinaison de tâches dont le résultat sert de point de départ pour la phase suivante, et ainsi de suite. Les tâches constituant chaque phase sont itératives et exigent une analyse et une conception minutieuses afin que les résultats de la phase

concernée soient fiables. Il se peut que les premières phases du déploiement doivent elles aussi être répétées. Par exemple, au cours de la phase de conception du déploiement, vous vous rendez compte que l'analyse de la phase précédente n'est pas satisfaisante et qu'elle doit être réexaminée.

Les sections suivantes fournissent une description succincte de chaque phase du cycle de vie de la solution.

Phase d'analyse d'exploitation

Au cours de cette phase, vous définissez les objectifs d'exploitation de votre projet de déploiement et formulez les exigences d'entreprise devant être satisfaites pour que ces objectifs soient atteints. Lors de la définition des exigences de l'entreprise, tenez compte des contraintes d'exploitation susceptibles d'avoir un impact sur la réalisation des objectifs d'exploitation. Tout au long du cycle de vie, vous évaluez la planification de votre déploiement et, en dernier lieu, la solution de déploiement obtenue en fonction des analyses effectuées au cours de la phase d'analyse d'exploitation.

Lors de la phase d'analyse d'exploitation, vous créez des documents d'exigences d'entreprise sur lesquels vous vous appuyerez pour passer à la phase de spécification technique.

Pour plus d'informations sur la phase d'analyse d'exploitation, reportez-vous au [Chapitre 2](#).

Phase de spécification technique

Cette phase se fonde sur les exigences de l'entreprise et les contraintes d'exploitation définies au cours de la phase d'analyse d'exploitation. Elle consiste à les convertir en spécifications techniques, ensuite utilisées lors de la conception de l'architecture du déploiement. Les exigences techniques précisent des caractéristiques de qualité de service, notamment en termes de performances, de disponibilité et de sécurité.

Au cours de la phase de spécification technique, vous créez des documents contenant les informations suivantes :

- analyse des tâches utilisateur et des modèles d'utilisation ;
- cas d'utilisation conditionnant l'interaction utilisateur avec le système planifié ;
- exigences de qualité de service découlant des exigences de l'entreprise et prenant éventuellement en compte l'analyse des tâches utilisateur et des types d'utilisation.

Les documents d'analyse d'utilisation, de cas d'utilisation et d'exigences de qualité de service obtenus servent de point de départ pour la phase de conception logique du cycle de vie de la solution. L'analyse d'utilisation joue également un rôle important lors de la phase de conception du déploiement.

Au cours de la phase de spécification technique, vous pouvez également définir les exigences de niveau de service qui permettront de créer les contrats de niveau de service (SLA, Service Level Agreement). Ceux-ci précisent les conditions dans lesquelles la maintenance du système doit faire l'objet d'un support client. Ils sont généralement signés lors de l'approbation du projet, qui intervient au cours de la phase de conception du déploiement.

Pour plus d'informations sur les exigences techniques, reportez-vous au [Chapitre 3](#).

Phase de conception logique

Au cours de cette phase, vous identifiez les composants de Java Enterprise System requis pour implémenter une solution à l'aide des cas d'utilisation identifiés lors de la phase de spécification technique. Il vous faut également déterminer les composants fournissant la prise en charge des composants Java ES requis, ainsi que les composants personnalisés nécessaires pour satisfaire les exigences de l'entreprise. Vous mappez ensuite ces composants au sein d'une architecture logique décrivant les relations existant entre tous les composants. L'architecture logique ne décrit pas les composants matériels requis pour implémenter la solution.

Le résultat de la phase de conception logique est l'architecture logique. Seule, elle ne suffit pas pour passer à la phase de conception du déploiement. Les exigences de qualité de service établies lors de la phase de spécification technique sont également indispensables. L'architecture logique et les exigences de qualité de service issues de la phase de spécification technique constituent un scénario de déploiement. Ce scénario sert d'amorce à la phase de conception du déploiement.

Pour plus d'informations sur l'architecture logique, reportez-vous au [Chapitre 4](#).

Phase de conception du déploiement

Cette phase consiste à mapper les composants de l'architecture logique dans un environnement physique afin de définir une architecture de déploiement de haut niveau. Au cours de cette phase, vous créez également des spécifications d'implémentation précisant les informations de bas niveau concernant la création de l'architecture de déploiement. Enfin, des plans et des spécifications détaillant différents aspects de l'implémentation de la solution logicielle sont également élaborés.

L'approbation du projet intervient au cours de cette phase. Le coût du déploiement est évalué au moment de l'approbation du projet. Si celui-ci est approuvé, il faut procéder à la signature des contrats relatifs à l'implémentation du déploiement et se procurer les ressources destinées à la mise en œuvre du projet. L'approbation intervient généralement après que les spécifications d'implémentation ont été définies en détail. Elle peut également se produire une fois l'architecture du déploiement terminée.

Les résultats de la phase de conception du déploiement sont les suivants :

- **Architecture de déploiement** : document de conception de haut niveau représentant le mappage des composants sur le matériel et les logiciels du réseau.
- **Spécifications d'implémentation** : spécifications détaillées servant à la planification du déploiement.
- **Plans d'implémentation** : groupe de plans et de spécifications couvrant divers aspects de l'implémentation d'une solution logicielle d'entreprise. Ils comprennent des plans de migration, d'installation, de gestion des utilisateurs, de test, etc.

Pour plus d'informations sur la conception du déploiement, reportez-vous au [Chapitre 5](#).

Phase d'implémentation

Au cours de cette phase, vous partez des spécifications et des plans créés au cours de la conception du déploiement pour mettre en place l'architecture du déploiement et implémenter la solution. Selon la nature de votre projet de déploiement, cette phase implique l'exécution de certaines des tâches suivantes ou de la totalité d'entre elles :

- installation et configuration de l'infrastructure matérielle ;
- installation et configuration du logiciel ;
- modélisation des utilisateurs et des ressources au sein d'une conception d'annuaire LDAP ;
- migration des données à partir des répertoires et des bases de données existants conformément à un plan de gestion des utilisateurs ;
- création et déploiement de pilotes et de prototypes dans un environnement de test ;
- conception et exécution de tests fonctionnels destinés à évaluer la conformité avec les exigences système ;
- conception et exécution de tests de charge pour mesurer les performances du système en situation de charge de pointe ;
- développement et intégration des applications d'entreprise personnalisées ;
- création d'un déploiement de production pouvant être exécuté par phases.

Une fois un déploiement mis en production, vous pouvez passer à la phase de fonctionnement du cycle de vie de la solution.

Pour plus d'informations sur la phase d'implémentation, reportez-vous au [Chapitre 6](#).

Phase de fonctionnement

Cette phase comprend des tâches permettant d'implémenter le déploiement dans les meilleures conditions. Ces tâches sont les suivantes :

- contrôle du déploiement pour vérifier que le système s'exécute conformément aux prévisions ;
- réglage des performances pour permettre une exécution optimale des logiciels déployés ;
- exécution des opérations de maintenance planifiées lors d'un fonctionnement normal et des opérations non planifiées le cas échéant ;
- mise à niveau des logiciels et du matériel selon les besoins.

Notez que ce manuel n'a pas pour objet de décrire de façon détaillée la phase de fonctionnement.

Analyse d'exploitation

Dans le cycle de vie de la solution, la phase d'analyse d'exploitation consiste à définir des objectifs d'exploitation en analysant un problème donné et en identifiant les exigences de l'entreprise et les contraintes d'exploitation liées à la réalisation de ces objectifs.

Ce chapitre se compose des sections suivantes :

- "À propos de l'analyse d'exploitation" à la page 27
- "Définition des exigences de l'entreprise" à la page 28
- "Définition des contraintes d'exploitation" à la page 33

À propos de l'analyse d'exploitation

L'analyse d'exploitation commence par l'évaluation des objectifs d'exploitation. Vous devez ensuite analyser les problèmes d'exploitation à résoudre et identifier les conditions requises pour atteindre les objectifs fixés. Vous devez également prendre en compte les contraintes d'exploitation qui risquent de compromettre la réalisation des objectifs. L'analyse des exigences et des contraintes d'exploitation débouche sur la création de documents d'exigences d'exploitation.

Vous utiliserez ces documents au cours de la phase de définition des exigences techniques. Tout au long du cycle de vie de la solution, vous évalueriez la réussite de la planification de votre déploiement et, finalement, celle de votre solution en fonction des analyses effectuées au cours de la phase d'analyse d'exploitation.

Définition des exigences de l'entreprise

Il n'existe aucune formule simple permettant d'identifier toutes les exigences de l'entreprise. Celles-ci sont déterminées en collaboration avec les demandeurs de la solution logicielle, en fonction de votre connaissance du domaine d'activité et de façon réfléchie et créative.

Cette section comporte un certain nombre de facteurs à prendre en compte lors de la définition des exigences de l'entreprise.

Définition des objectifs d'exploitation

L'analyse d'exploitation permet de formuler les objectifs d'un projet de déploiement. L'établissement d'objectifs précis permet de prendre des décisions appropriées et d'éviter toute dérive du projet. La comparaison des objectifs d'exploitation avec les opérations en cours constitue également une aide à la prise de décisions.

Étendue du projet

Les exigences de l'entreprise doivent préciser l'étendue du projet de déploiement. Veillez à identifier les problèmes pouvant être résolus et évitez de définir des exigences trop floues qui pourraient nuire à la compréhension et à la réalisation des objectifs. Un cadre mal défini peut déboucher sur des problèmes de déploiement, que la conception de ce dernier soit inadaptée aux besoins de l'entreprise ou trop exigeante en ressources.

Priorités

Classez vos objectifs par ordre de priorité de sorte que les aspects les plus importants du déploiement soient pris en compte en premier. Des ressources limitées peuvent nécessiter un report ou une modification de certains objectifs. Par exemple, certains déploiements, en raison de leur ampleur et de leur complexité, exigent une implémentation progressive de la solution. La définition de priorités constitue une aide à la prise des décisions qui détermineront l'acceptation de votre projet de déploiement par les parties prenantes.

Qualités essentielles

Identifiez les éléments clés de la réussite de votre projet afin de permettre aux demandeurs et aux concepteurs de la solution de se concentrer sur les critères les plus importants.

Facteurs de croissance

Lors de la définition des objectifs d'exploitation, prenez en compte non seulement les besoins actuels de l'entreprise, mais également la façon dont ils peuvent évoluer et s'accroître au fil du temps. À défaut, votre solution serait vite dépassée.

Marge de sécurité

La conception de votre solution se fonde sur des hypothèses formulées au cours de la phase d'analyse d'exploitation. Il est possible que celles-ci ne soient pas pertinentes, pour diverses raisons (données insuffisantes, erreurs de jugement ou imprévus). Veillez à garder une marge de sécurité qui s'appliquera à la fois à vos objectifs d'exploitation et à la phase de planification, et qui permettra à votre solution de faire face aux impondérables.

Identification des besoins des utilisateurs

Effectuez les recherches nécessaires pour connaître les destinataires de la solution, leurs besoins et les bénéfices auxquels ils s'attendent. Vous pouvez, par exemple, classer les utilisateurs selon les catégories suivantes :

- employés actuels uniquement ;
- employés actuels et anciens employés ;
- administrateurs ;
- clients actifs ;
- tous les clients ;
- site sur lequel l'utilisateur travaille ;
- public général ;
- accès restreint.

Formulez clairement les attentes des utilisateurs afin de prendre les bonnes décisions lors de la conception de la solution. Par exemple, une solution peut apporter les avantages suivants aux utilisateurs :

- accès distant aux ressources de l'entreprise ;
- collaboration au sein de l'entreprise ;
- simplification des tâches quotidiennes ;
- partage des ressources par des équipes distantes ;
- productivité accrue ;
- autoadministration par les utilisateurs finals.

Développement des exigences opérationnelles

Définissez les exigences opérationnelles comme un ensemble d'exigences fonctionnelles axées sur des objectifs précis. Généralement, les spécifications opérationnelles s'appliquent aux éléments suivants :

- fonctionnalité pour l'utilisateur final ;
- réduction des temps de réponse ;
- disponibilité et temps de disponibilité ;
- réduction des taux d'erreurs ;
- archivage et conservation des informations.

Exprimez ces exigences opérationnelles en termes clairs accessibles à toutes les parties prenantes du projet. Évitez les termes ambigus tels « temps de réponse utilisateur approprié ». Voici quelques exemples de formulation d'exigences opérationnelles :

- reprise du fonctionnement normal dans les 10 minutes suivant une panne ;
- possibilité de récupérer les messages reçus au cours des 48 dernières heures ;
- exécution des transactions en ligne en 60 secondes maximum au cours des périodes d'activité intense ;
- authentification des utilisateurs finals en moins de 4 secondes au cours des périodes d'activité intense.

Prise en charge des types d'utilisation existants

Exprimez les types d'utilisation existants sous forme d'objectifs mesurables. Pour déterminer ces objectifs, posez-vous les questions suivantes :

- Comment les services actuels sont-ils utilisés ?
- Quels est le type d'utilisation (occasionnelle, fréquente ou intense) ?
- Quels sont les sites auxquels les utilisateurs se connectent le plus fréquemment ?
- Quelle est la taille moyenne des messages envoyés ?
- Combien de transactions les utilisateurs réalisent-ils par jour ou par heure ?

Intéressez-vous aux utilisateurs qui font appel à vos services. Pour définir vos objectifs, identifiez par exemple la fréquence et la durée d'utilisation des services existants. Si votre expérience dans le domaine ne vous permet pas d'établir ces types d'utilisation, étudiez ceux en vigueur dans des entreprises similaires à la vôtre.

Connaissance de la culture de l'entreprise

L'analyse des exigences doit tenir compte de divers aspects de la culture et de la politique de l'entreprise. Le non-respect de la culture de l'entreprise peut aboutir à une mauvaise compréhension de la solution ou à des difficultés d'implémentation.

Parties prenantes

Identifiez les personnes et les organisations les plus concernés par la réussite de la solution proposée. Toutes les parties prenantes doivent participer activement à la définition des exigences et des objectifs d'exploitation. Pour éviter tout défaut de

conception important, elles doivent également être informées des modifications planifiées. Une partie prenante insuffisamment impliquée ou mal informée pourrait même porter préjudice au bon déroulement de l'implémentation du déploiement.

Normes et stratégies

Une bonne compréhension des normes et des stratégies est indispensable à l'entreprise demandeuse de la solution. En effet, ces éléments peuvent affecter les aspects techniques du projet, le choix des produits et la méthode de déploiement.

La confidentialité des données du personnel, détenues et contrôlées par le service des ressources humaines ou les responsables des services, peut être un exemple de norme à prendre en compte. Les procédures appliquées par l'entreprise pour la gestion des modifications en sont un autre exemple. Les stratégies de gestion des modifications peuvent en effet avoir des incidences sur l'acceptation de la solution, ainsi que sur la méthode et le planning d'implémentation.

Exigences réglementaires

Ces exigences varient de manière importante selon le type d'entreprise concerné. Soyez attentif aux exigences réglementaires qui peuvent avoir des répercussions sur le déploiement. De nombreuses entreprises et organismes gouvernementaux exigent une conformité parfaite aux normes d'accessibilité. Lors du déploiement de solutions globales, renseignez-vous sur les lois et les réglementations en vigueur dans les pays étrangers concernés. Par exemple, dans la plupart des pays européens, le stockage des informations personnelles est très contrôlé.

Sécurité

Certains de vos objectifs peuvent soulever des problèmes de sécurité qu'il est important d'identifier. Mettez en évidence les objectifs de sécurité dont dépend le succès de la solution. Par exemple :

- accès autorisé aux informations exclusives ;
- accès basé sur des rôles pour les informations confidentielles ;
- communication sécurisée entre les sites distants ;
- lancement des applications distantes sur les systèmes locaux ;
- transactions sécurisées avec des entreprises et des organisations tierces ;
- application des stratégies de sécurité.

Répartition géographique

La répartition géographique des sites et la bande passante entre ces derniers peuvent avoir une incidence sur les décisions à prendre en termes de conception. Certains sites peuvent nécessiter une gestion locale.

Ces considérations géographiques peuvent avoir de nombreuses répercussions sur le projet : coûts de formation en hausse, complexité accrue, etc. Identifiez clairement les exigences liées à la répartition géographique des sites. Mettez en évidence les sites jouant un rôle clé dans le fonctionnement de la solution.

Approche incrémentielle

En règle générale, les solutions logicielles sont considérées comme des systèmes complets. Or, la plupart du temps, leur déploiement s'effectue de manière incrémentielle, par étapes.

L'approche incrémentielle implique la création d'une feuille de route définissant les étapes qui aboutiront à la mise en œuvre de la solution complète. Vous devrez également envisager des plans à court terme pour les aspects de la solution dont l'implémentation a été reportée à une date ultérieure.

L'approche incrémentielle permet de bénéficier des avantages suivants :

- adaptation aux modifications des exigences dues au développement de l'entreprise ;
- possibilité d'exploiter l'infrastructure existante lors de la phase d'implémentation du déploiement ;
- respect des exigences liées aux dépenses d'investissement ;
- mobilisation réduite de personnel ;
- possibilité d'envisager des partenariats.

Contrats de niveau de service (SLA)

Un contrat de niveau de service (SLA, Service Level Agreement) indique les performances minimales requises et, au cas où ces exigences ne seraient pas satisfaites, le niveau et l'étendue du support client devant être fourni. Il est fondé sur les exigences de l'entreprise définies au cours de l'analyse d'exploitation, qui deviennent des exigences de niveau de service au cours de la phase de spécification technique. Le SLA est signé lors de l'approbation du projet, laquelle intervient au cours de la phase de conception du déploiement.

Un SLA doit être articulé autour de notions telles que les temps de disponibilité, de réponse, de livraison des messages et de reprise sur sinistre. Il doit contenir une description du système, présenter les rôles et les responsabilités des organismes de support, expliquer comment mesurer les niveaux de service, décrire les demandes de modification, etc. Pour déterminer l'étendue d'un SLA, vous devez définir les attentes de l'entreprise en matière de disponibilité du système.

Définition des contraintes d'exploitation

Ces contraintes jouent un rôle important dans la définition du type de projet de déploiement. Un déploiement réussi dépend essentiellement de la réalisation des objectifs d'exploitation dans le respect des contraintes d'exploitation. Celles-ci peuvent être d'ordre financier, physique (la capacité du réseau, par exemple), temporel (date butoir correspondant à un événement important tel la prochaine réunion annuelle) ou toute autre contrainte susceptible d'affecter la réalisation des objectifs d'exploitation.

Cette section décrit un certain nombre de facteurs à prendre en considération lors de la définition des contraintes d'exploitation.

Problèmes de migration

Souvent, un projet de déploiement vient en remplacement ou en complément d'une infrastructure logicielle et de données existantes. La migration des données et des procédures de l'infrastructure existante vers la nouvelle solution doit être prise en compte, tout comme l'interopérabilité avec les applications existantes. Une analyse de l'infrastructure actuelle est nécessaire pour déterminer la portée des problèmes de migration liés à la nouvelle solution.

Établissement de la planification

La planification de l'implémentation d'une solution peut avoir des répercussions sur les décisions de conception. Une planification trop ambitieuse peut entraîner une révision à la baisse des objectifs, une modification des priorités ou l'application d'une approche incrémentielle. Certains jalons importants d'une planification peuvent également exiger une attention particulière. Ces jalons peuvent être déterminés par des événements internes (ex. : changements de service planifiés) ou externes (ex. : début d'un trimestre scolaire).

Limitations budgétaires

Les projets de déploiement doivent respecter un certain budget. L'évaluation du coût de mise en place de la solution proposée et des ressources requises pour l'exploiter tout au long de sa durée de vie repose sur les éléments suivants :

- **Matériel et infrastructure de réseau existants** : l'exploitation de l'infrastructure existante peut affecter la conception d'un système.
- **Ressources de développement requises pour implémenter la solution** : lorsque les ressources sur le plan matériel, logiciel et humain sont limitées, il est préférable de s'orienter vers un déploiement incrémentiel. Vous devrez peut-être faire appel

aux mêmes ressources ou équipes de développement pour chaque phase incrémentielle implémentée.

- **Maintenance, administration et support** : analysez les ressources disponibles dans ces trois domaines. Des ressources limitées peuvent avoir un impact sur les décisions de conception.

Coût de possession

Outre la maintenance, l'administration et le support, analysez les autres facteurs intervenant dans le coût de possession. Les mises à niveau matérielles et logicielles requises, l'impact de la solution sur le réseau électrique, le coût des télécommunications, etc. sont autant de facteurs pesant sur les frais courants. Les SLA spécifiant des niveaux de disponibilité ont aussi un impact sur le coût de possession dans la mesure où ils exigent une redondance accrue.

L'implémentation d'une solution elle-même doit produire un retour sur investissement. L'analyse du retour sur investissement consiste à estimer les bénéfices générés par rapport aux capitaux investis.

L'évaluation des bénéfices financiers d'une solution implique l'analyse attentive des objectifs visés par rapport à ceux atteints par d'autres solutions, ou au fait de ne rien changer.

Exigences techniques

Dans le cycle de vie de la solution, la phase de spécification technique consiste à effectuer une analyse d'utilisation, identifier les cas d'utilisation et déterminer les exigences de qualité de service pour la solution de déploiement proposée.

Ce chapitre se compose des sections suivantes :

- "À propos des exigences techniques" à la page 35
- "Analyse d'utilisation" à la page 36
- "Cas d'utilisation" à la page 38
- "Exigences de qualité de service" à la page 39
- "Exigences de niveau de service" à la page 47

À propos des exigences techniques

L'analyse des exigences techniques commence par un examen des exigences de l'entreprise exposées lors de la phase d'analyse d'exploitation. À l'aide de l'analyse d'exploitation, vous devez :

- effectuer une analyse d'utilisation afin de connaître les conditions de charge escomptées ;
- créer des cas d'utilisation constituant un modèle d'interaction utilisateur classique avec le système ;
- créer un ensemble d'exigences de qualité de service définissant le comportement d'une solution déployée en termes de temps de réponse, de disponibilité, de sécurité, etc.

Les exigences de qualité de service dépendent de l'analyse d'utilisation et des cas d'utilisation, tout en tenant compte des exigences et des contraintes d'entreprise identifiées précédemment.

Les exigences de qualité de service sont ensuite associées aux architectures logiques au cours de la phase de conception logique afin de créer un scénario de déploiement. Ce scénario joue un rôle capital au cours de la phase de conception du déploiement dans le cycle de vie de la solution.

À l’instar de l’analyse d’exploitation, il n’existe aucune formule simple pour l’analyse des exigences techniques qui permette de générer l’analyse d’utilisation, les cas d’utilisation ou la configuration système requise. L’analyse des exigences techniques nécessite une connaissance du domaine d’activité, des objectifs d’entreprise et de la technologie sous-jacente du système.

Analyse d’utilisation

L’analyse d’utilisation implique l’identification des divers utilisateurs de la solution en cours de conception ainsi que la définition des types d’utilisation. Ces informations servent de point de départ pour évaluer les conditions de charge sur le système. Les informations relatives à l’analyse d’utilisation sont également utiles lors de l’assignation de pondérations aux cas d’utilisation, comme décrit dans la section “Cas d’utilisation” à la page 38.

Au cours de l’analyse d’utilisation, vous devez consulter les utilisateurs chaque fois que vous en avez la possibilité, rechercher les données sur les types d’utilisation existants et interroger les concepteurs et administrateurs des systèmes précédents. Le tableau suivant répertorie les paramètres à prendre en considération lors d’une analyse d’utilisation.

TABLEAU 3-1 Facteurs à prendre en compte lors de l’analyse d’utilisation

Facteur	Description
Nombre et type d’utilisateurs	Identifiez le nombre d’utilisateurs que la solution doit prendre en charge et classez-les, si nécessaire. Par exemple : <ul style="list-style-type: none">■ Une solution B2C (Business to Customer) peut comporter un grand nombre de visiteurs, mais seul un petit nombre d’entre eux s’enregistre et s’engage dans des transactions commerciales.■ Une solution B2E (Business to Employee) doit prendre en compte chaque employé, sachant que certains employés peuvent avoir besoin d’accéder au réseau interne à partir de l’extérieur. Dans une solution de ce type, il se peut que les dirigeants aient besoin d’une autorisation pour pouvoir accéder à certaines pages inaccessibles au reste des employés.

TABLEAU 3-1 Facteurs à prendre en compte lors de l'analyse d'utilisation (Suite)

Facteur	Description
Utilisateurs actifs et inactifs	<p>Identifiez les types d'utilisation et les rapports entre les utilisateurs actifs et inactifs.</p> <p>Les utilisateurs actifs sont des utilisateurs connectés au système qui interagissent avec les services du système. Les utilisateurs inactifs peuvent être des utilisateurs qui ne sont pas connectés, des utilisateurs connectés mais qui n'interagissent pas avec les composants du système ou des utilisateurs appartenant à la base de données mais qui ne se connectent jamais.</p>
Utilisateurs administratifs	<p>Identifiez les utilisateurs qui doivent accéder au système pour le contrôler, le mettre à jour et prendre en charge son déploiement.</p> <p>Déterminez les types d'utilisation administratifs spécifiques ayant un impact sur les exigences techniques (par exemple, l'administration du déploiement en dehors du pare-feu).</p>
Types d'utilisation	<p>Identifiez la façon dont les différents types d'utilisateurs accèdent au système et fixez les objectifs d'utilisation attendus.</p> <p>Par exemple :</p> <ul style="list-style-type: none">■ Existe-t-il des périodes de pointe en matière d'utilisation ?■ Quelles sont les heures de bureau classiques ?■ Les utilisateurs sont-ils répartis de façon homogène ?■ Quelle est la durée moyenne d'une connexion utilisateur ?
Augmentation du nombre d'utilisateurs	<p>Déterminez si le nombre d'utilisateurs est fixe ou si le déploiement implique une augmentation du nombre d'utilisateurs.</p> <p>Essayez d'évaluer de façon raisonnable cet éventuel accroissement.</p>
Transactions utilisateur	<p>Identifiez le type de transactions utilisateur à prendre en charge. Ces transactions peuvent être converties en cas d'utilisation.</p> <p>Par exemple :</p> <ul style="list-style-type: none">■ Quelles sont les tâches effectuées par les utilisateurs ?■ Une fois connectés, les utilisateurs le restent-ils ? En général, effectuent-ils quelques tâches avant de se déconnecter ?■ La collaboration entre les utilisateurs est-elle importante au point de rendre indispensable l'utilisation de calendriers communs, la tenue de conférences Web et le déploiement de pages Web internes ?

TABLEAU 3-1 Facteurs à prendre en compte lors de l'analyse d'utilisation (Suite)

Facteur	Description
Études d'utilisateurs et données statistiques	<p>Utilisez les études d'utilisateurs ainsi que d'autres éléments existants pour déterminer les types de comportement de l'utilisateur.</p> <p>Souvent, les entreprises ou les organisations industrielles effectuent des études sur les utilisateurs à partir desquelles il est possible d'extraire des informations intéressantes. Les fichiers journaux des applications existantes peuvent contenir des données statistiques utiles à l'élaboration de projections pour un système.</p>

Cas d'utilisation

Les cas d'utilisation simulent une interaction utilisateur classique avec le système en cours de conception et décrivent le déroulement complet d'une opération du point de vue d'un utilisateur final. En matière de conception, un classement par ordre de priorité autour d'un ensemble complet de cas d'utilisation assure la livraison des fonctionnalités prévues. Les cas d'utilisation jouent un rôle capital dans la conception logique.

Assignez des pondérations relatives aux cas d'utilisation, sachant que les pondérations les plus élevées correspondent aux tâches utilisateur les plus fréquentes. Lors de la conception, le système de pondération des cas d'utilisation permet de déterminer les services système les plus utilisés.

Les cas d'utilisation peuvent être classés selon deux niveaux :

- **Rapports sur les cas d'utilisation** : descriptions de chaque cas d'utilisation comprenant les flux d'événements principaux et secondaires.
- **Diagrammes des cas d'utilisation** : diagrammes illustrant les relations entre les acteurs et les cas d'utilisation, présentant ainsi une organisation plus formelle du flux des événements. Les diagrammes des cas d'utilisation permettent de créer des cas d'utilisation étendus ou complexes. Pour dessiner les diagrammes relatifs aux cas d'utilisation, la norme généralement utilisée est la norme UML (Unified Modeling Language).

Exigences de qualité de service

Les exigences de qualité de service (QoS) sont des spécifications techniques indiquant les qualités système de fonctions telles les performances, la disponibilité, l'évolutivité et l'entretien. Ces éléments sont fonction des besoins de l'entreprise spécifiés dans les exigences de l'entreprise. Par exemple, si des services doivent être disponibles 24 heures sur 24 toute l'année, l'exigence de disponibilité doit être établie en fonction.

Le tableau ci-dessous répertorie les qualités système constituant la base des exigences de qualité de service.

TABEAU 3-2 Qualités système ayant un impact sur les exigences de qualité de service

Qualité du système	Description
Performances	Mesure du temps de réponse et de la capacité de traitement par rapport aux conditions de charge utilisateur.
Disponibilité	Mesure de la fréquence à laquelle les ressources et services d'un système sont accessibles aux utilisateurs finals, également appelée <i>temps de disponibilité</i> du système.
Évolutivité	Possibilité d'ajouter de la capacité (et des utilisateurs) à un système déployé au fil du 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.
Sécurité	Combinaison complexe de facteurs décrivant l'intégrité d'un système et de ses utilisateurs. La sécurité comprend l'authentification et l'autorisation des utilisateurs, la sécurité des données ainsi que l'accès sécurisé à un système.
Capacité latente	Aptitude d'un système à traiter des charges de pointe inhabituelles sans ressources supplémentaires. La capacité latente est un paramètre important des qualités d'évolutivité, de performances et de disponibilité.
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.

Les qualités d'un système sont étroitement liées. Les exigences liées à un critère de qualité système peuvent avoir un impact sur les exigences et la conception d'autres critères de qualité système. 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 de serveurs en vue de pallier les problèmes de disponibilité peut avoir des conséquences sur l'entretien (coûts de maintenance).

Il est essentiel de comprendre la manière dont les qualités système sont liées et de savoir quels sont les compromis nécessaires à la conception d'un système répondant à la fois aux exigences et aux contraintes de l'entreprise.

Les sections suivantes décrivent en détail les qualités système ayant un impact sur la conception du déploiement et fournissent des informations sur les facteurs à prendre en considération lorsque vous répertoriez les exigences de qualité de service. Vous trouverez également une section consacrée aux exigences de niveau de service sur lesquels sont fondés les contrats de niveau de service.

Performances

Dans les exigences de l'entreprise, les performances (temps de réponse) sont généralement exprimées en termes non techniques. Voici un exemple d'exigence d'entreprise relative à l'accès Web :

Lors de la connexion, les utilisateurs s'attendent à un temps de réponse raisonnable, à savoir, pas plus de quatre secondes.

En partant de cette exigence, examinez la totalité des cas d'utilisation afin de savoir comment exprimer cette exigence au niveau du système. Dans certains cas, il se peut que vous souhaitiez inclure les conditions de charge utilisateur définies au cours de l'analyse d'utilisation. Indiquez ensuite l'exigence de performances pour chaque cas d'utilisation en termes de temps de réponse par rapport aux conditions de charge indiquées ou de temps de réponse ajouté à la capacité de traitement. Il vous est également possible d'indiquer le nombre d'erreurs autorisées.

Voici deux exemples de formulation des exigences système en termes de performances :

- Le temps de réponse correspondant à l'actualisation des pages Web, échantillonné toutes les 15 minutes, ne doit dépasser à aucun moment de la journée les quatre secondes et ne doit pas comporter plus de 3,4 erreurs par million de transactions.
- En période de pointe, tout utilisateur doit pouvoir établir 25 connexions sécurisées par seconde avec un temps de réponse ne dépassant pas 12 secondes, le nombre d'erreurs par million de transactions ne devant pas dépasser 3,4.

Les exigences de performances sont étroitement liées aux exigences de disponibilité (influence du basculement sur les performances) et à la capacité latente (aptitude à traiter des charges de pointe inhabituelles).

Disponibilité

Il s'agit du temps de disponibilité du système. La disponibilité s'exprime en pourcentage de temps pendant lequel le système est accessible aux utilisateurs. Une indisponibilité, c'est-à-dire, le temps pendant lequel le système n'est pas accessible, peut provenir d'une défaillance matérielle ou logicielle, d'une défaillance réseau ou de

toute autre anomalie (par exemple, une coupure d'alimentation) engendrant une panne système. Une interruption planifiée de l'activité en vue d'effectuer des opérations d'entretien (maintenance et mises à niveau) n'est pas considérée comme une indisponibilité en tant que telle. Voici une équation élémentaire permettant de calculer la disponibilité en termes de pourcentage de temps d'accessibilité du système :

$$\text{Disponibilité} = \frac{\text{temps de disponibilité}}{(\text{temps de disponibilité} + \text{temps d'indisponibilité})} * 100 \%$$

En règle générale, la disponibilité est fonction du nombre de « neuf » obtenu. Par exemple, une disponibilité de 99 % correspond à deux neuf. Lorsque vous indiquez davantage de neuf, la conception du déploiement s'en trouve fortement modifiée. Le tableau suivant indique le temps d'indisponibilité non planifié, correspondant à chaque neuf ajouté, dans le cadre d'un système fonctionnant 24 heures sur 24, 7 jours sur 7 et ce, toute l'année (soit un total de 8 760 heures).

TABLEAU 3-3 Interruption d'activité non planifiée d'un système fonctionnant toute l'année (8 760 heures)

Nombre de neuf	Pourcentage disponible	Indisponibilité non planifiée
2	99 %	88 heures
3	99,9 %	9 heures
4	99,99 %	45 minutes
5	99,999 %	5 minutes

Systèmes à tolérance de pannes

Dans le cadre d'une disponibilité de quatre à cinq neuf, le système doit être doté d'une tolérance de pannes. Un système à tolérance de pannes doit continuer de fonctionner même en cas de défaillance matérielle ou logicielle. La tolérance de pannes est généralement obtenue par une redondance matérielle (CPU, mémoire et périphériques réseau) ou logicielle permettant d'assurer les services essentiels.

Un point de panne unique correspond à un composant logiciel ou matériel qui fait partie d'un chemin critique mais qui n'est pas sauvegardé par des composants redondants. La panne de ce composant entraîne la perte de service pour le système. Lors de la conception d'un système à tolérance de pannes, vous devez identifier et supprimer les points de pannes uniques potentiels.

Ces systèmes peuvent s'avérer coûteux à implémenter et à entretenir. Il est nécessaire de comprendre la nature des exigences de l'entreprise concernant la disponibilité et de tenir compte des stratégies et des coûts des solutions de disponibilité qui répondent à ces exigences.

Classement par ordre de priorité en termes de disponibilité de service

D'un point de vue utilisateur, la disponibilité concerne certains services en particulier plutôt que la totalité du système. Par exemple, la non-disponibilité d'un service de messagerie instantanée a généralement peu ou pas d'effet sur la disponibilité d'autres services. En revanche, la non-disponibilité d'un service dont dépendent plusieurs autres services (notamment Directory Server) a un impact plus important. Les spécifications relatives à une disponibilité élevée doivent clairement faire référence aux cas d'utilisation spécifiques et aux analyses d'utilisation nécessitant une disponibilité accrue.

Il peut s'avérer utile de répertorier les besoins de disponibilité selon un ensemble ordonné de priorités. Le tableau suivant classe par ordre de priorité la disponibilité de divers types de services.

TABLEAU 3-4 Disponibilité des services par ordre de priorité

Priorité	Type de service	Description
1	Services essentiels	Il s'agit des services qui doivent être disponibles en permanence. Par exemple, les services de bases de données (tels les annuaires LDAP) pour les applications.
2	Disponibilité indispensable	Il s'agit des services qui doivent rester disponibles, mais dont les performances peuvent être réduites. Par exemple, il se peut que la disponibilité du service de messagerie ne soit pas fondamentale dans certains environnements de travail.
3	Possibilité de différer	Il s'agit de services dont la nécessité de disponibilité est limitée dans le temps. Par exemple, il se peut que la disponibilité des services de calendrier ne soit pas essentielle dans certains environnements de travail.
4	Facultatif	Il s'agit des services qui peuvent être différés sans limite de temps. Par exemple, dans certains environnements, les services de messagerie instantanée peuvent être considérés comme utiles mais pas obligatoires.

Panne de services

La conception de disponibilité prend également en considération ce qui se passe lorsque la disponibilité est compromise ou lorsqu'un composant est défectueux. Autrement dit, si un utilisateur connecté doit ou non redémarrer sa session et si une panne risque d'affecter d'autres zones du système. Les exigences de qualité de service doivent tenir compte de ces scénarios et indiquer comment le déploiement doit réagir face à ces situations.

Évolutivité

L'évolutivité correspond à la possibilité d'ajouter de la capacité à un système de sorte qu'il puisse accepter une charge supplémentaire provenant d'utilisateurs existants ou d'un accroissement de leur nombre. L'évolutivité requiert généralement davantage de ressources mais ne doit pas entraîner de modifications au niveau de la conception de l'architecture du déploiement ni de perte de service due au temps utilisé pour l'ajout de ressources.

Comme la disponibilité, l'évolutivité concerne certains services en particulier plutôt que la totalité du système. En revanche, pour les services dont dépendent d'autres services, tels Directory Server, l'évolutivité peut avoir un impact sur l'ensemble du système.

Il n'est pas nécessaire de mentionner les exigences d'évolutivité dans les exigences de qualité de service, sauf si l'extension du déploiement est clairement définie dans les exigences de l'entreprise. Cependant, lors de la phase de conception du déploiement de la solution, l'architecture de déploiement doit toujours prévoir une certaine tolérance pour la mise à l'échelle du système et ce, même dans le cas où aucune exigence en matière d'évolutivité n'a été indiquée.

Estimation du développement

L'estimation du développement d'un système en vue de déterminer les exigences d'évolutivité implique l'utilisation de projections et d'estimations aléatoires. Vous devez tenir compte des trois éléments ci-dessous pour dresser la liste des exigences liées à un système évolutif.

- **Stratégie de conception hautes performances** : au cours de la spécification des exigences de performances, indiquez une capacité latente afin de gérer les charges pouvant augmenter au fil du temps. Optimisez également la disponibilité en tenant compte des contraintes budgétaires. Cette stratégie vous permet de gérer le développement et de mieux planifier les jalons de la mise à l'échelle du système.
- **Déploiement incrémentiel** : le déploiement incrémentiel facilite la planification de l'ajout de ressources. Indiquez précisément chaque jalon de la mise à l'échelle du système. On appelle jalons les exigences de charge fixées à l'avance pour évaluer l'évolutivité.
- **Contrôle complet de performance** : le contrôle des performances permet de déterminer le moment où des ressources doivent être ajoutées au système. Les exigences en termes de contrôle des performances sont autant d'indications pour les opérateurs et administrateurs responsables de la maintenance et des mises à niveau.

Le tableau suivant répertorie les facteurs à prendre en considération pour la détermination des exigences d'évolutivité.

TABLEAU 3-5 Facteurs d'évolutivité

Facteur	Description
Analyse des types d'utilisation	En examinant les données existantes, vous pouvez comprendre les types d'utilisation des utilisateurs actuels ou prévus. En l'absence de telles données, analysez les données du secteur économique concerné ou les estimations du marché.
Conception pour une échelle maximale acceptable	Conception ayant pour objectif l'échelle maximale requise pour satisfaire les demandes connues et les demandes possibles. Il s'agit généralement d'une estimation s'étalant sur 24 mois basée sur l'évaluation des performances de la charge utilisateur existante et des estimations acceptables en matière de charge à venir. La durée de l'estimation dépend fortement de la fiabilité des projections.
Définition de jalons appropriés	Implémentez la conception du déploiement par incrément afin de répondre aux exigences à court terme et utilisez pour cela une mémoire tampon qui puisse faire face à une éventuelle croissance. En outre, vous devez définir des jalons pour l'ajout de ressources système. Par exemple : <ul style="list-style-type: none">■ Achat important (trimestriel ou annuel)■ Délai d'obtention de matériel et de logiciel (1 à 6 semaines)■ Mémoire tampon (10 à 100 %, selon les prévisions de croissance)
Intégration de nouvelles technologies	Il est nécessaire de s'imprégner des nouvelles technologies, telles que les processeurs et les serveurs Web plus rapides. Il faut également connaître la façon dont elles affectent les performances de l'architecture sous-jacente.

Exigences liées à la sécurité

La sécurité est un élément complexe qui intervient à tous les niveaux d'un système déployé. L'objectif principal du développement des exigences liées à la sécurité consiste à identifier les menaces de sécurité et à développer une stratégie pour les combattre. Cette analyse de sécurité se compose des étapes suivantes :

1. Identification des éléments critiques
2. Identification des menaces pesant sur ces éléments
3. Identification des failles de sécurité présentant un risque pour l'organisation
4. Développement d'un plan de sécurité afin de limiter les risques spécifiques à votre organisation

L'analyse des exigences de sécurité nécessite une collaboration de la part des parties prenantes de votre organisation, notamment les directeurs, les analystes d'exploitation et le personnel du service des technologies de l'information. Il arrive très souvent qu'une organisation désigne une personne responsable de la sécurité et chargée de la conception et de l'implémentation des mesures de sécurité.

La section suivante décrit quelques-uns des domaines abordés en matière de planification de la sécurité.

Éléments d'un plan de sécurité

La planification de la sécurité d'un système fait partie de la conception du déploiement et est essentielle à la réussite de l'implémentation. Vous devez tenir compte des points suivants lors de la planification de la sécurité :

- **Sécurité physique** : la sécurité physique concerne l'accès physique aux routeurs, aux serveurs, aux salles de serveurs, aux centres de données et à d'autres parties de votre infrastructure. Toutes les mesures de sécurité peuvent se révéler inutiles si une personne non autorisée pénètre dans la salle des serveurs et déconnecte les routeurs.
- **Sécurité réseau** : la sécurité réseau concerne l'accès à votre réseau par l'intermédiaire de pare-feux, les zones d'accès sécurisées, les listes de contrôle d'accès, ainsi que l'accès au port. Dans le cadre de la sécurité réseau, vous développez des stratégies contre les accès non autorisés, les falsifications et les attaques de refus de service.
- **Sécurité des applications et des données d'application** : la sécurité des applications et des données d'application traite de l'accès aux comptes utilisateur, aux données d'entreprise et aux applications d'entreprise par le biais de procédures et de stratégies d'authentification et d'autorisation. Les stratégies suivantes doivent être définies :
 - stratégies de mot de passe ;
 - droits d'accès, tels que l'administration déléguée aux utilisateurs par opposition à l'accès administrateur ;
 - inactivation du compte ;
 - contrôle d'accès ;
 - stratégies de chiffrement, y compris le transport sécurisé de données et l'utilisation de certificats pour la signature de données.
- **Sécurité des personnes** : la stratégie de sécurité d'une organisation définit l'environnement de travail et les exercices pratiques auxquels les utilisateurs doivent se conformer pour garantir l'exécution des autres mesures de sécurité. Il est généralement préférable de concevoir un manuel de sécurité et d'offrir une formation aux utilisateurs sur les exercices de sécurité. Des exercices de sécurité fiables doivent faire partie de la culture de votre organisation pour assurer une stratégie de sécurité globale efficace.

Capacité latente

La capacité latente correspond à l'aptitude d'un déploiement à faire face aux pics de charge inhabituels sans utiliser de ressources supplémentaires. Généralement, il n'est pas nécessaire de spécifier les exigences de qualité de service directement liées à la capacité latente. Cependant, cette dernière représente un facteur important en termes de disponibilité, de performances et d'évolutivité du système.

Exigences d'entretien

L'entretien est la facilité avec laquelle un système déployé peut être entretenu, englobant des tâches telles la surveillance du système, la résolution des problèmes, l'ajout et la suppression d'utilisateurs et la mise à niveau des composants matériels et logiciels.

Lors de la planification des exigences d'entretien, tenez compte des facteurs répertoriés dans le tableau suivant :

TABLEAU 3-6 Facteurs concernant les exigences d'entretien

Facteur	Description
Planification de l'interruption d'activité	<p>Identification des tâches de maintenance entraînant l'indisponibilité totale ou partielle de services spécifiques.</p> <p>Certaines opérations de maintenance et de mise à niveau peuvent être invisibles pour l'utilisateur, tandis que d'autres nécessitent une interruption de service. Lorsque cela est possible, planifiez avec les utilisateurs les activités de maintenance nécessitant une indisponibilité de service afin qu'ils puissent s'organiser en fonction.</p>
Types d'utilisation	<p>Identification des types d'utilisation en vue déterminer le moment le plus approprié pour la planification des opérations de maintenance.</p> <p>Par exemple, programmez les opérations de maintenance en soirée ou en week-end pour les systèmes sur lesquels les pics d'utilisation se produisent pendant les heures de bureau classiques. Pour les systèmes distribués géographiquement, cette programmation peut s'avérer plus compliquée à mettre en place.</p>

TABLEAU 3-6 Facteurs concernant les exigences d'entretien (Suite)

Facteur	Description
Disponibilité	<p>L'entretien reflète généralement votre conception de disponibilité. Les stratégies visant à limiter les interruptions d'activité pour la maintenance et les mises à niveau sont liées à votre stratégie de disponibilité. Les systèmes ayant besoin d'un degré élevé de disponibilité ne disposent que de peu de temps pour la maintenance, les mises à niveau et les réparations.</p> <p>Les stratégies de gestion des exigences de disponibilité se répercutent sur la manière dont vous gérez la maintenance et les mises à niveau. Par exemple, sur des systèmes distribués géographiquement, l'entretien peut dépendre de la possibilité d'acheminer des charges de travail vers des serveurs distants pendant les opérations de maintenance.</p> <p>De plus, les systèmes nécessitant un degré de disponibilité élevé peuvent exiger la mise en place de solutions plus complexes automatisant au maximum le redémarrage des systèmes.</p>
Diagnostics et contrôle	<p>Vous pouvez améliorer la stabilité du système en exécutant régulièrement des outils de diagnostic et de contrôle pour identifier les zones problématiques.</p> <p>Le contrôle régulier d'un système permet d'éviter les problèmes, de gérer des charges de travail selon les stratégies de disponibilité et d'améliorer la planification de la maintenance et des indisponibilités.</p>

Exigences de niveau de service

Un contrat de niveau de service (SLA, Service Level Agreement) indique les performances minimales requises et, au cas où ces exigences ne seraient pas satisfaites, le niveau et l'étendue du support client devant être fourni. Les exigences de niveau de service correspondent aux exigences système indiquant les conditions du contrat.

Comme pour les exigences de qualité de service, les exigences de niveau de service reposent sur les exigences d'entreprise et constituent une garantie de qualité globale que le système déployé doit respecter. Les spécifications concernant les exigences de niveau de service doivent être claires, car il s'agit d'un contrat. Les exigences de niveau de service définissent avec exactitude les conditions sous lesquelles les exigences sont testées et définissent également les raisons pour lesquelles ces exigences ne sont pas satisfaites.

Conception logique

La phase de conception logique du cycle de vie de la solution consiste à définir une architecture logique qui décrit les relations existant entre les composants logiques de la solution. Cette architecture, associée à l'analyse d'utilisation établie lors de la phase de spécification technique, constitue un scénario de déploiement, qui sert de base pour la phase de conception du déploiement.

Ce chapitre se compose des sections suivantes :

- "À propos des architectures logiques" à la page 49
- "Conception d'une architecture logique" à la page 51
- "Composants Java Enterprise System" à la page 51
- "Exemple d'architecture logique" à la page 58
- "Zones d'accès" à la page 67
- "Scénario de déploiement" à la page 68

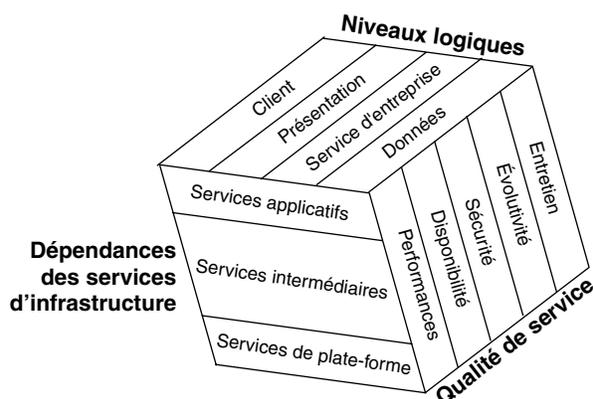
À propos des architectures logiques

L'architecture logique identifie les composants logiciels nécessaires à l'implémentation d'une solution et décrit les relations existant entre ces composants. L'architecture logique et les exigences de qualité de service définies lors de la phase de spécification technique constituent un scénario de déploiement. Celui-ci sert de base pour concevoir l'architecture de déploiement intervenant au cours de la phase suivante, à savoir la conception du déploiement.

Lorsque vous développez une architecture logique, vous devez identifier non seulement les composants qui fournissent des services aux utilisateurs, mais également ceux qui fournissent les logiciels intermédiaires et les services de plateforme requis. Vous pouvez également identifier les dépendances des services d'infrastructure et les niveaux logiques.

Les dépendances des services d'infrastructure et les niveaux logiques sont deux des trois dimensions de l'architecture de la solution sur lesquelles Sun Java™ Enterprise System repose. Ces trois dimensions sont répertoriées ci-dessous et représentées à la section "À propos des architectures logiques" à la page 49.

- **Dépendances des services d'infrastructure** : composants logiciels en interaction qui fournissent des services d'entreprise. Ces composants nécessitent un ensemble sous-jacent de services d'infrastructure qui permettent aux composants distribués de communiquer les uns avec les autres et d'interagir.
- **Niveaux logiques** : organisation logique des composants logiciels par niveaux représentant l'indépendance logique et physique des composants logiciels selon la nature des services qu'ils fournissent.
- **Qualité de service** : qualités de service système, telles les performances, la disponibilité, l'évolutivité et d'autres qualités illustrant divers aspects de la conception et du fonctionnement d'une solution logicielle.



Remarque – Pour plus d'informations sur les concepts d'architecture de Java Enterprise System, consultez le chapitre "Java Enterprise System Architecture" du manuel *Présentation technique de Sun Java Enterprise System 2005Q4*.

Une architecture logique illustre les niveaux de service d'infrastructure en décrivant les composants nécessaires et leurs dépendances. Elle répartit également les composants sur des niveaux logiques qui correspondent aux services de présentation, d'entreprise et de données accessibles au niveau client. Les exigences de qualité de service ne sont pas modélisées dans l'architecture logique, mais associées à cette dernière dans un scénario de déploiement.

Conception d'une architecture logique

Lors de la conception d'une architecture logique, tenez compte des cas d'utilisation identifiés au cours de la phase de spécification technique pour déterminer les composants de Java Enterprise System qui fournissent les services requis par la solution. Vous devez également déterminer les composants qui fournissent des services aux composants préalablement identifiés.

Les composants de Java Enterprise System sont positionnés dans une architecture à plusieurs niveaux en fonction du type de service qu'ils fournissent. Cette approche vous permettra de définir la manière dont les services fournis par les composants doivent être distribués et d'élaborer une stratégie d'implémentation des exigences de qualité de service (évolutivité, disponibilité, etc.)

Vous pouvez également répartir les composants logiques dans des zones d'accès sécurisé. À la section "[Zones d'accès](#)" à la page 67, vous trouverez des exemples de ces zones.

Composants Java Enterprise System

Java Enterprise System est constitué de composants logiciels en interaction fournissant des services d'entreprise à utiliser pour créer votre solution. Le schéma ci-dessous illustre les principaux composants logiciels de Java Enterprise System. Le manuel *Présentation technique de Sun Java Enterprise System 2005Q4* fournit des informations complémentaires sur les composants Java Enterprise System et leurs services.

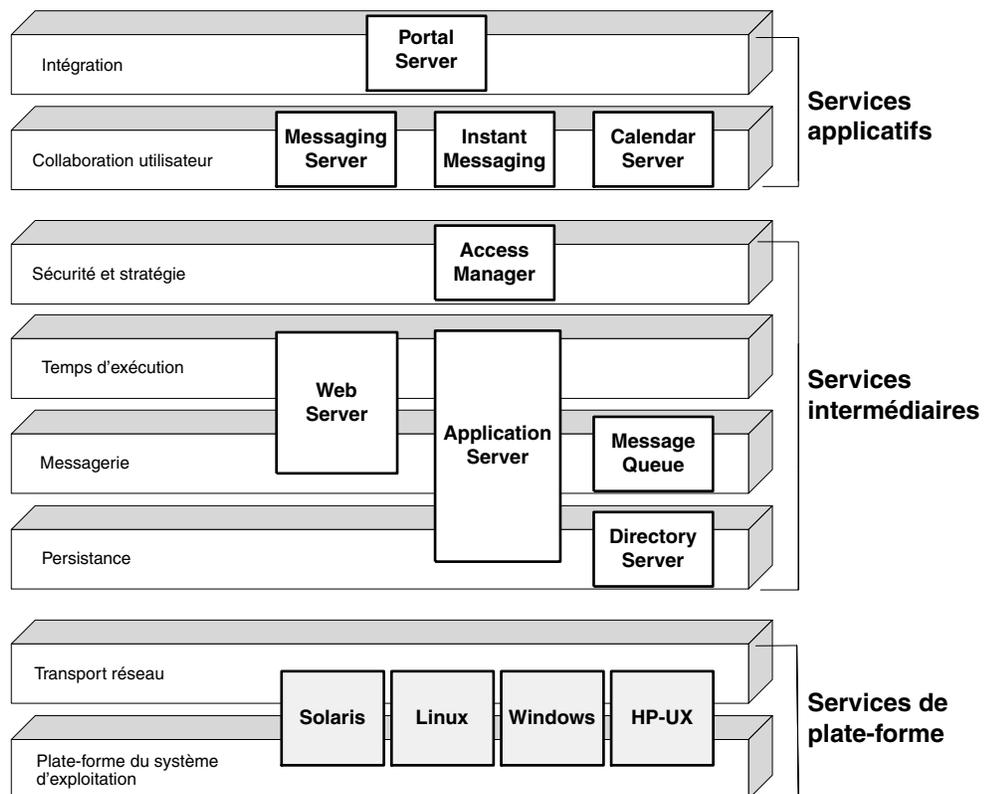


FIGURE 4-1 Composants Java Enterprise System

Dépendances entre composants

Lorsque vous identifiez les composants Java Enterprise System d'une architecture logique, vous devez également identifier les composants qui les prennent en charge. Par exemple, si vous identifiez Messaging Server en tant que composant requis d'une architecture logique, cette architecture doit également comprendre Directory Server et, éventuellement, Access Manager. En effet, Messaging Server dépend de Directory Server pour les services d'annuaire et d'Access Manager pour les solutions exigeant une connexion unique.

Le tableau ci-dessous répertorie les dépendances entre les composants de Java Enterprise System. Reportez-vous à la section "[Dépendances entre composants](#)" à la page 52 pour une représentation visuelle des dépendances entre les composants clés. Lors de la conception d'une architecture logique, utilisez ce tableau et le schéma associé pour identifier les composants dépendants dans votre conception.

TABLEAU 4-1 Dépendances entre composants de Java Enterprise System

Composant Java Enterprise System	Dépendant de
Application Server	Message Queue ; Directory Server (facultatif)
Calendar Server	Messaging Server (pour le service de notification par e-mail) ; Access Manager (pour la connexion unique) ; Web Server (pour l'interface Web) ; Directory Server
Communications Express	Access Manager (pour la connexion unique) ; Calendar Server Messaging Server ; Instant Messaging ; Web Server (pour l'interface Web) ; Directory Server
Directory Proxy Server	Directory Server
Directory Server	Aucune
Access Manager	Application Server ou Web Server ; Directory Server
Instant Messaging	Access Manager (pour la connexion unique) ; Directory Server
Message Queue	Directory Server (facultatif)
Messaging Server	Access Manager (pour la connexion unique) ; Web Server (pour l'interface Web) ; Directory Server
Portal Server	S'il est configuré en vue de l'utilisation de canaux Portal Server : Calendar Server ; Messaging Server ; Instant Messaging Access Manager (pour la connexion unique) ; Application Server ou Web Server ; Directory Server
Portal Server Secure Remote Access	Portal Server
Web Server	Access Manager (facultatif, pour le contrôle d'accès)

Remarque – La liste des dépendances entre composants de Java Enterprise System figurant à la section “[Dépendances entre composants](#)” à la page 52 n’est pas exhaustive. En effet, les dépendances que vous devez prendre en compte lors de la planification de l’installation n’y sont pas répertoriées. Pour une liste complète des dépendances de composants Java Enterprise System, reportez-vous au manuel *Guide d’installation de Sun Java Enterprise System 2005Q4 pour UNIX*.

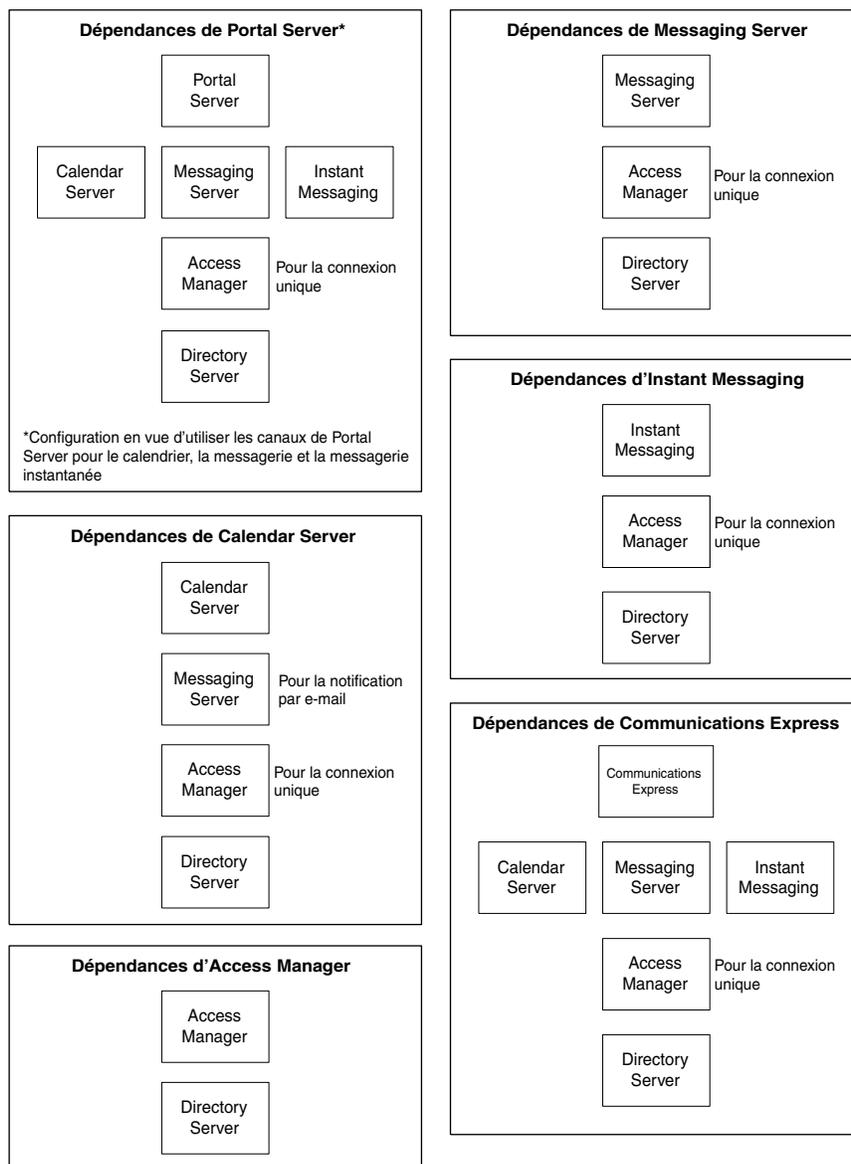


FIGURE 4-2 Dépendances entre composants de Java Enterprise System

Prise en charge d'un conteneur Web

La section “[Dépendances entre composants](#)” à la page 52 précédente ne fait pas état du conteneur Web au sein duquel s'exécutent Portal Server et Access Manager. Ce conteneur peut être fourni par Application Server, Web Server ou un produit tiers. Si votre architecture logique inclut Portal Server ou Access Manager, veillez à utiliser le conteneur Web nécessaire à ces composants.

Services logiques distincts fournis par Messaging Server

Le composant Messaging Server de Java Enterprise System peut être configuré pour exécuter des instances séparées fournissant les services logiques distincts suivants :

- Message Transfer Agent
- Message Multiplexor
- Message Express Multiplexor
- Message Store

Ces différentes configurations de Messaging Server offrent des fonctionnalités pouvant être déployées sur des serveurs physiques distincts et être présentes à différents niveaux d'une architecture logique. Étant donné que ces configurations de Messaging Server correspondent à des services logiques distincts sur différents niveaux, elles doivent être considérées comme des composants logiques distincts dans l'architecture logique. La section “[Exemple d'architecture logique](#)” à la page 58 propose un exemple de composants logiques distincts.

Le tableau ci-dessous décrit les configurations logiques distinctes de Messaging Server.

TABLEAU 4-2 Configurations de Messaging Server

Sous-composant	Description
Message Transfer Agent (MTA)	Prend en charge l'envoi de messages en traitant les connexions SMTP, en acheminant les messages et en les livrant aux mémoires de messages appropriées. Les composants MTA peuvent être configurés pour la prise en charge des messages provenant de l'extérieur (entrants) ou envoyés de l'entreprise (sortants).
Message Store (STR)	Permet la récupération et le stockage des messages.
Message Multiplexor (MMP)	Prend en charge la récupération des messages en accédant aux mémoires de messages des clients de messagerie à l'aide du protocole IMAP ou POP.
Messenger Express Multiplexor (MEM)	Prend en charge la récupération des messages en accédant aux mémoires de messages pour le compte de clients Web (HTTP).

Composants d'accès

Java Enterprise System fournit également des composants qui assurent un accès aux services système, souvent depuis des sites situés hors du pare-feu de l'entreprise. Certaines configurations de Messaging Server peuvent également fournir un accès réseau lorsque Messaging Server est configuré pour le multiplexeur de messages, par exemple). Le tableau ci-dessous répertorie les composants de Java Enterprise System qui assurent un accès distant aux services système.

TABLEAU 4-3 Composants de Java Enterprise System fournissant un accès distant

Composant	Description
Directory Proxy Server	Fournit des services améliorés d'accès aux annuaires, de compatibilité de schéma, de routage et d'équilibrage de charge pour des plusieurs instances de Directory Server.
Portal Server, Portal Server Secure Remote Access	Fournit un accès Internet sécurisé de l'extérieur d'un pare-feu d'entreprise au contenu et aux services de Portal Server, y compris les portails internes et les applications Internet.
Portal Server, Portal Server Mobile Access	Fournit un accès sans fil depuis des périphériques mobiles et un accès vocal à Portal Server.
Messaging Server Message Multiplexor (MMP)	Prend en charge la récupération des messages en accédant aux mémoires de messages pour le compte de clients Web (HTTP).

Les composants fournissant un accès distant sont généralement déployés dans des zones d'accès sécurisé, comme l'illustre l'exemple de la section "[Zones d'accès](#)" à la page 67.

Conception d'architecture à plusieurs niveaux

Java Enterprise System est adapté à la conception d'architectures à plusieurs niveaux, où les services sont répartis selon les fonctionnalités qu'ils proposent. Chaque service est logiquement indépendant et accessible par des services situés au même niveau ou sur un autre niveau. Le schéma ci-dessous décrit un modèle d'architecture à plusieurs niveaux pour des applications d'entreprise, modèle comportant les niveaux client, présentation, services d'entreprise et données.

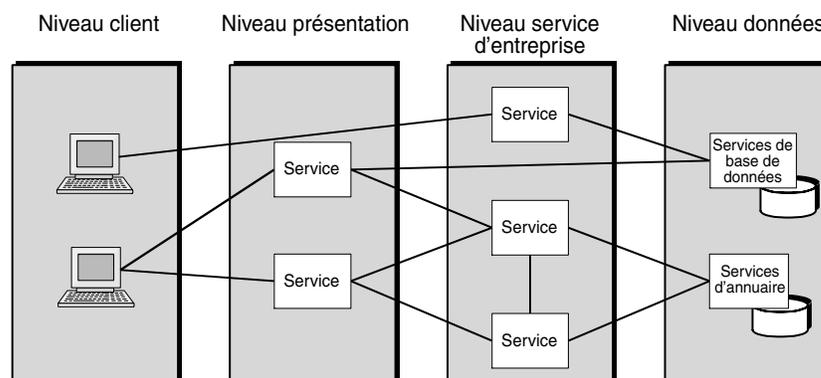


FIGURE 4-3 Modèle d'architecture à plusieurs niveaux

Le tableau suivant fait un récapitulatif des niveaux logiques décrits à la section "Conception d'architecture à plusieurs niveaux" à la page 56.

TABLEAU 4-4 Niveaux logiques dans une architecture à plusieurs niveaux

Niveau	Description
Niveau client	Contient les applications clientes qui présentent des informations aux utilisateurs finals. Dans le cas de Java Enterprise System, il s'agit généralement de clients de messagerie, de navigateurs Web ou de clients d'accès mobile.
Niveau présentation	Fournit les services qui présentent des données aux utilisateurs finals et permet à ceux-ci de traiter et de manipuler la présentation de ces données. Par exemple, un client de messagerie Web ou le composant Portal Server permettent aux utilisateurs de modifier la présentation des informations qu'ils reçoivent.
Niveau services d'entreprise	Fournit des services d'arrière-plan chargés de récupérer les données du niveau correspondant et de les fournir aux autres services des niveaux présentation ou services d'entreprise, ou directement aux clients du niveau client. Par exemple, Access Manager fournit des services d'identité aux autres composants de Java Enterprise System.
Niveau données	Fournit des services de base de données aux services des niveaux présentation ou services d'entreprise. Par exemple, Directory Server fournit un accès à l'annuaire LDAP pour les autres services.

Une conception d'architecture à plusieurs niveaux offre plusieurs avantages. Au cours de la phase de conception du déploiement, la répartition des services selon leur fonctionnalité vous permet de déterminer la façon dont vous allez les distribuer au sein de votre réseau. De plus, vous pouvez voir comment les composants de l'architecture accèdent aux services des autres composants. Cette transparence vous permet de planifier les solutions de qualité de service en termes de disponibilité, d'évolutivité, de sécurité, etc.

Exemple d'architecture logique

Dans cette section, vous trouverez des exemples d'architectures logiques pour des solutions Java Enterprise System. Ces exemples indiquent comment répartir des composants logiques sur les niveaux appropriés de l'architecture et comment analyser des relations entre composants en examinant des cas d'utilisation. Utilisez les exemples d'architectures logiques de cette section pour mieux comprendre la conception d'architectures logiques dans les solutions Java Enterprise System.

Le premier exemple correspond à une solution de Messaging Server élémentaire illustrant les interactions entre les composants logiques distincts de Messaging Server et les autres composants. Le second exemple montre l'architecture logique d'une solution de déploiement basée sur les identités pouvant convenir à une entreprise de taille moyenne (de 1 000 à 5 000 employés).

Exemple de Messaging Server

Le schéma ci-dessous illustre une architecture logique de base pour un déploiement de Messaging Server. Cette architecture comporte uniquement les composants logiques distincts requis pour Messaging Server. Les relations entre ces composants sont présentées dans les schémas apparaissant plus bas.

Remarque – Généralement, un déploiement de Messaging Server fait partie d'une solution d'entreprise incluant d'autres composants de Java Enterprise System, comme indiqué à la section ["Exemple de communications basées sur les identités"](#) à la page 63.

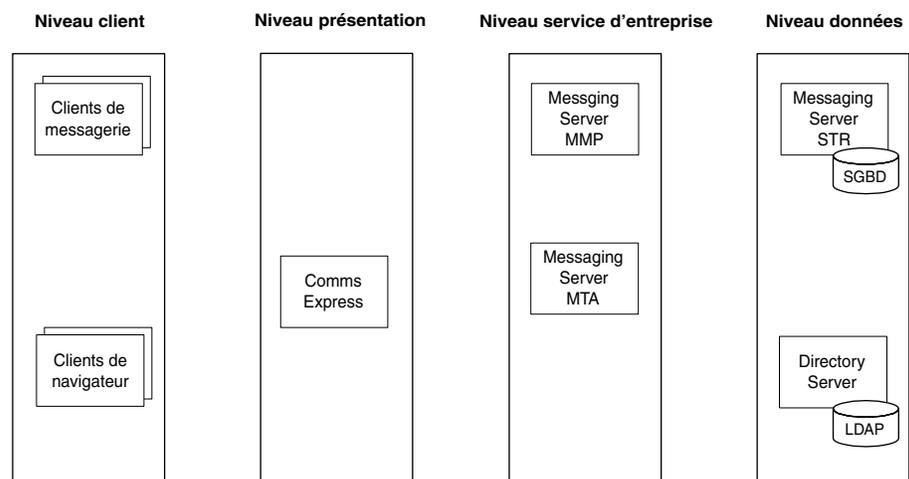


FIGURE 4-4 Architecture logique d'un déploiement Messaging Server

Le tableau ci-dessous décrit les composants présentés dans la section "Exemple de Messaging Server" à la page 58.

TABLEAU 4-5 Composants de l'architecture logique de Messaging Server

Composant	Description
Clients de messagerie	Applications clientes pour la lecture et l'envoi de messages.
Messaging Server MTA	Configuration de Messaging Server en tant qu'agent de transfert de messages (MTA) pour recevoir, acheminer, transporter et livrer des messages.
Messaging Server MMP	Configuration de Messaging Server en tant que multiplexeur de messages (MMP) en vue d'acheminer les connexions vers les mémoires de messages appropriées pour la récupération et le stockage. MMP accède à Directory Server pour rechercher les informations d'annuaire afin d'identifier la mémoire de messages adéquate.
Messaging Server STR	Configuration de Messaging Server en tant que mémoire de messages pour la récupération et le stockage de messages.
Directory Server	Fournit l'accès aux données d'annuaire LDAP.

L'architecture ne définit pas la répllication des services pour les composants de Messaging Server. En général, dans les déploiements d'entreprise, des instances de MTA entrant et sortant distinctes sont créées mais seul un composant MTA apparaît dans le schéma de l'"Exemple de Messaging Server" à la page 58. La décision de répliquer les composants logiques en plusieurs instances se prend lors de la phase de conception du déploiement.

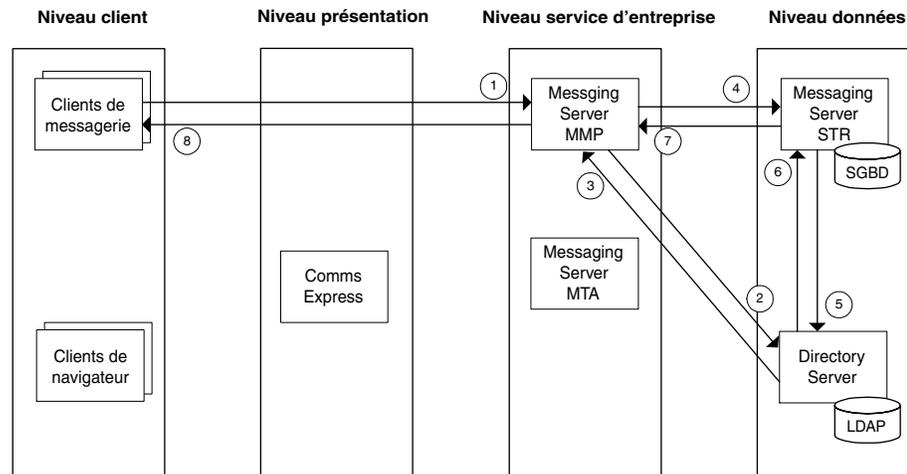
Cas d'utilisation de Messaging Server

Les cas d'utilisation permettent d'identifier les relations entre les composants logiques d'une architecture. En mappant les interactions entre composants en fonction des cas d'utilisation, vous obtenez une représentation visuelle de ces interactions s'avérant très utile pour la conception du déploiement.

Avant de commencer la conception du déploiement, il est préférable d'analyser des cas d'utilisation pour déterminer les interactions entre composants. Les trois cas d'utilisation ci-dessous, propres à Messaging Server, illustrent les interactions entre composants logiques.

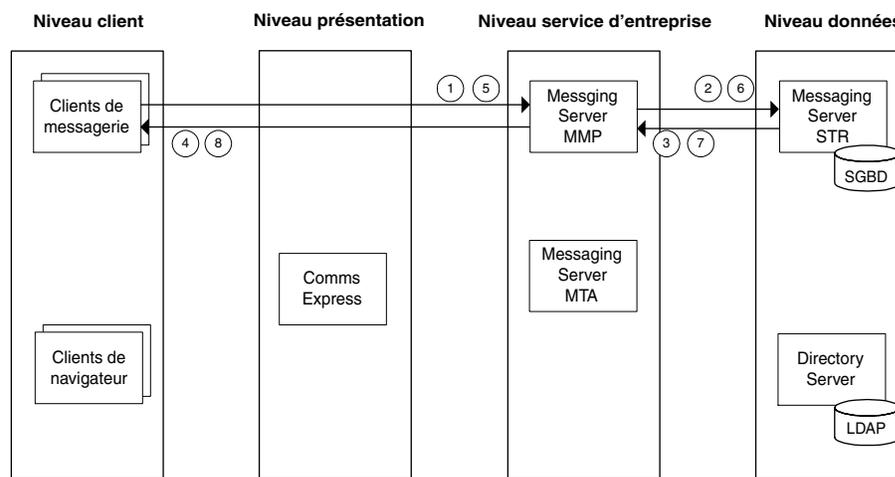
▼ Cas d'utilisation n°1 : connexion utilisateur à Messaging Server réussie

- Étapes**
1. Le client de messagerie transmet les informations de connexion à Messaging Server Multiplexor (MMP).
 2. MMP demande la vérification de l'ID utilisateur et du mot de passe à Directory Server.
 3. Directory Server renvoie la vérification à MMP.
 4. MMP demande la liste de messages à Messaging Server Message Store (STR).
 5. STR demande l'enregistrement LDAP de l'utilisateur à Directory Server.
 6. Directory Server renvoie l'enregistrement LDAP de l'utilisateur à STR.
 7. STR renvoie la liste de messages à MMP.
 8. MMP transfère la liste de messages au client de messagerie.



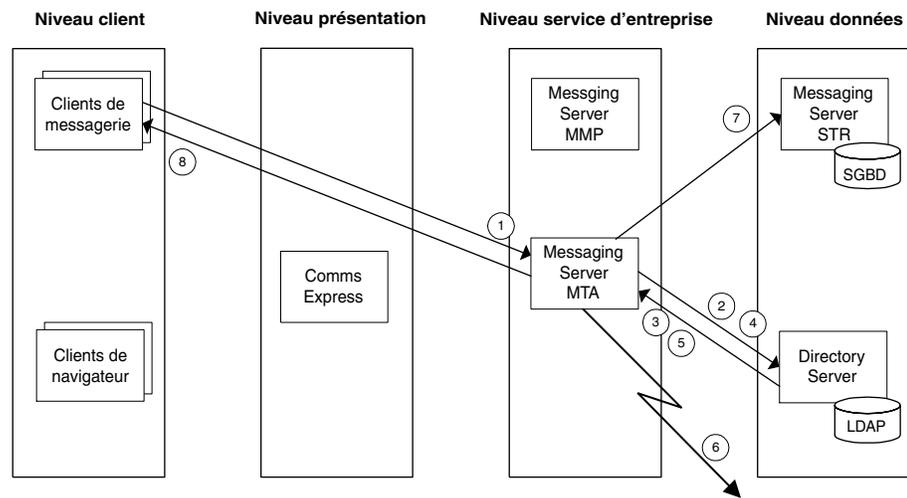
▼ Cas d'utilisation n°2 : lecture et suppression de messages par l'utilisateur connecté

- Étapes**
1. Le client de messagerie demande le message à lire à Messaging Server Multiplexor (MMP).
 2. MMP demande la liste de messages à Messaging Server Message Store (STR).
 3. STR renvoie le message à MMP.
 4. MMP transfère le message au client de messagerie.
 5. Le client de messagerie transmet l'action de suppression de message à MMP.
 6. MMP transfère l'action de suppression de message à STR.
 7. STR supprime le message de la base de données et en envoie la confirmation à MMP.
 8. MMP transfère la confirmation de suppression au client de messagerie.



▼ Cas d'utilisation n°3 : envoi d'un message par un utilisateur connecté

- Étapes**
1. Un message, composé dans le client de messagerie, est envoyé par ce dernier à Messaging Server Message Transfer Agent (MTA).
 2. MTA demande la vérification de l'ID utilisateur et du mot de passe à Directory Server.
 3. Directory Server renvoie la vérification au MTA.
 4. MTA vérifie le domaine de destination de chaque destinataire auprès de Directory Server.
 5. Directory Server renvoie le domaine de destination de chaque destinataire au MTA.
 6. MTA transfère le message à chaque destinataire.
 7. MTA achemine le message vers Messaging Server Message Store (STR) pour qu'il soit stocké dans la boîte d'envoi.
 8. MTA envoie la confirmation au client de messagerie.



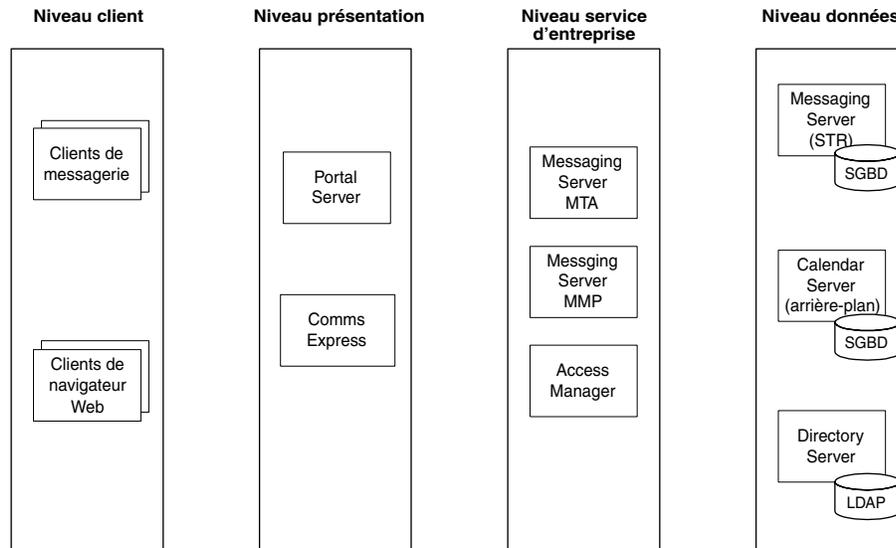
Exemple de communications basées sur les identités

Cet exemple illustre une solution de communications basées sur les identités pour une entreprise de taille moyenne (1 000 à 5 000 employés). Une analyse d'exploitation exhaustive, suivie d'une analyse détaillée des exigences techniques, est requise pour concevoir l'architecture logique. Cet exemple étant théorique, supposons que les exigences d'entreprise suivantes ont été définies :

- Les salariés de l'entreprise doivent disposer d'un accès personnalisé aux sites Web internes, aux services de communication, aux services de calendrier et à d'autres ressources.
- L'authentification et l'autorisation au niveau de l'entreprise fournissent l'accès aux sites Web internes et à d'autres services.
- Un suivi de toutes les identités uniques est effectué dans l'ensemble des services de l'entreprise, ce qui permet une connexion unique (Single Sign-On ou SSO) fournissant l'accès aux sites Web internes et à d'autres services.

Les cas d'utilisation associés à cet exemple détailleraient les procédures de connexion, la lecture et l'envoi de messages, la personnalisation du portail, la synchronisation des calendriers et d'autres activités utilisateur du même type.

Le schéma ci-dessous illustre une architecture logique pour ce type de solution de communications basées sur les identités.



Cas d'utilisation pour l'exemple de communications basées sur les identités

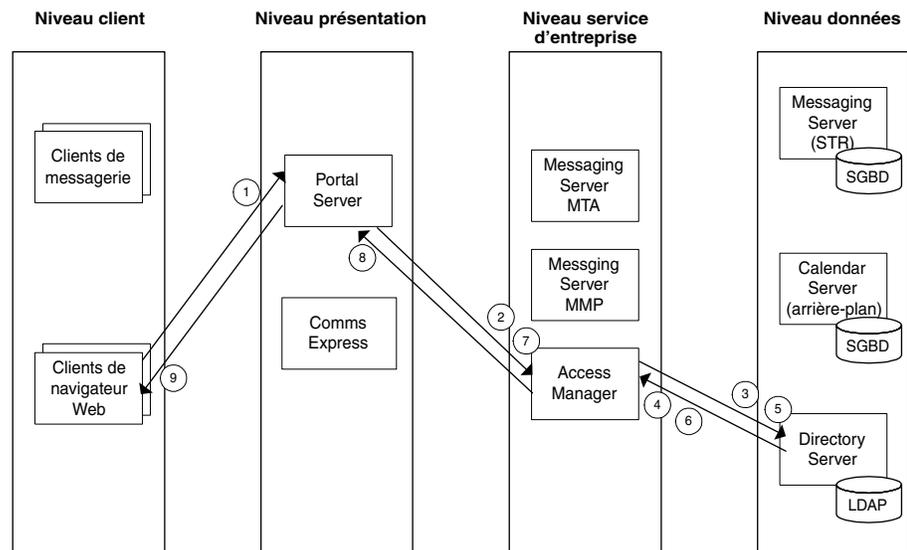
Pour une solution de déploiement de ce type, il existe de nombreux cas d'utilisation détaillés décrivant les interactions utilisateur avec les services fournis par la solution. Cet exemple détaille les interactions entre composants lors de la connexion d'un utilisateur à un portail à partir d'un client Web. Il divise ce scénario de connexion en deux cas d'utilisation :

- L'utilisateur se connecte, est authentifié, puis Portal Server récupère sa configuration de portail.
- Portal Server récupère les informations de messagerie et de calendrier afin de les afficher dans le client Web.

Ces deux cas d'utilisation peuvent être considérés comme un seul cas étendu. Toutefois, dans cet exemple, les cas d'utilisation ont été séparés dans un souci de simplification.

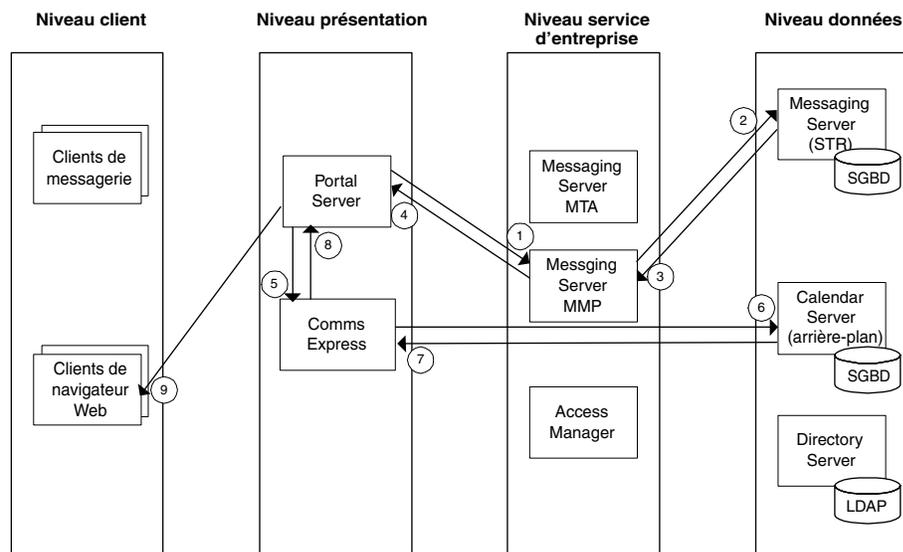
▼ Cas d'utilisation n°1 : l'utilisateur se connecte correctement et le portail récupère sa configuration

- Étapes**
1. Le client Web envoie l'ID utilisateur et le mot de passe à Portal Server.
 2. Portal Server fait une demande d'authentification auprès d'Access Manager.
 3. Access Manager demande la vérification de l'ID utilisateur et du mot de passe à Directory Server.
 4. Directory Server vérifie l'ID utilisateur et le mot de passe.
 5. Access Manager demande le profil utilisateur à Directory Server.
 6. Directory Server renvoie le profil utilisateur.
 7. Portal Server demande le profil d'affichage utilisateur à Access Manager.
 8. Access Manager renvoie la configuration du portail.
 9. La configuration du portail s'affiche dans le client de navigation Web.



▼ Cas d'utilisation n°2 : Portal Server affiche les informations de messagerie et de calendrier

- Étapes**
1. Une fois la connexion, l'authentification et la récupération de la configuration du portail correctement effectuées, Portal Server lance une requête pour récupérer les messages auprès de Messaging Server MMP.
 2. MMP demande la liste des messages à Messaging Server STR.
 3. STR renvoie la liste des messages à MMP.
 4. MMP transfère les en-têtes de messages à Portal Server.
 5. Portal Server lance une requête pour récupérer les informations de calendrier auprès de Communications Express.
 6. Communications Express lance une requête pour récupérer les informations de calendrier auprès du composant d'arrière-plan Calendar Server.
 7. Le composant d'arrière-plan Calendar Server renvoie les informations de calendrier à Communications Express.
 8. Communications Express transfère les informations de calendrier à Portal Server.
 9. Portal Server envoie les informations de canal au client de navigation Web.



Zones d'accès

Les composants d'une architecture logique peuvent également être placés dans des zones d'accès décrivant la façon dont l'architecture fournit un accès sécurisé. Le schéma suivant décrit les zones d'accès pour le déploiement. Chacune d'entre elles décrit la façon dont les composants fournissent un accès sécurisé à Internet et à l'intranet.

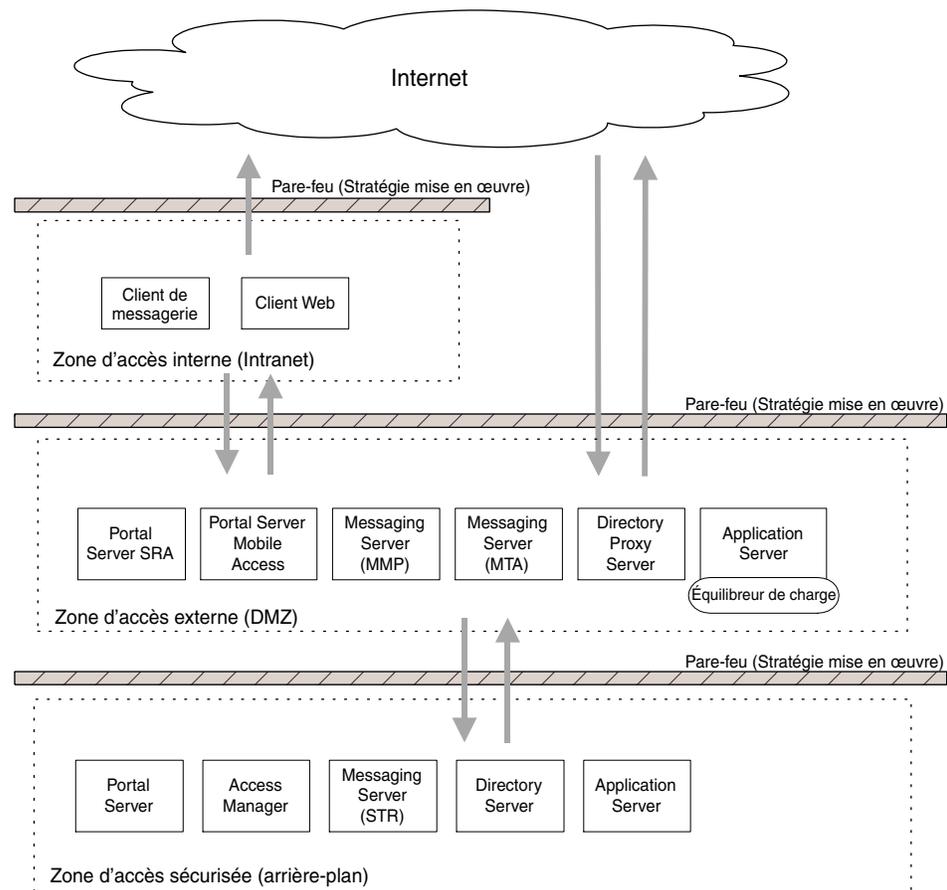


FIGURE 4-5 Composants logiques placés dans des zones d'accès

le tableau ci-dessous décrit les zones présentées dans la section consacrée aux "Zones d'accès" à la page 67.

TABLEAU 4-6 Zones d'accès sécurisé et leurs composants

Zone d'accès	Description
Zone d'accès interne (intranet) :	accès à Internet régi par des règles appliquées via un pare-feu entre l'intranet et Internet. La zone d'accès interne sert généralement aux utilisateurs finals pour la navigation sur le Web et l'envoi de messages. Dans certains cas, l'accès direct à Internet pour la navigation sur le Web est autorisé. Toutefois, l'accès sécurisé à et depuis Internet est généralement assuré via la zone d'accès externe.
Zone d'accès externe (DMZ)	Fournit un accès sécurisé à Internet en jouant le rôle de tampon de sécurité entre le Web et les services d'arrière-plan stratégiques.
Zone d'accès sécurisé (arrière-plan)	Fournit un accès restreint aux services d'arrière-plan stratégiques, accessibles uniquement par la zone d'accès externe.

Le schéma des "Zones d'accès" à la page 67 ne correspond pas aux niveaux logiques décrits dans les exemples précédents, mais fait apparaître en détail les composants qui fournissent un accès distant et interne, ainsi que les relations entre ces composants et les mesures de sécurité telles que les pare-feux. Il fournit une représentation visuelle des règles d'accès à appliquer. Combinez la conception d'architecture à plusieurs niveaux et celle des zones d'accès pour obtenir un modèle logique de votre déploiement planifié.

Scénario de déploiement

La conception de l'architecture de déploiement terminée ne suffit pas pour passer à la phase de conception du déploiement dans le cycle de vie de la solution. Vous devez l'associer aux exigences de qualité de service identifiées au cours de la phase de spécification technique. Pour constituer un scénario de déploiement valable, vous devez associer architecture logique et exigences de qualité de service. Ce scénario sert de point de départ pour concevoir l'architecture de déploiement, comme l'explique le [Chapitre 5](#).

Conception du déploiement

Au cours de la phase de conception du déploiement de la solution, vous préparez une architecture de déploiement de haut niveau et une spécification d'implémentation de bas niveau, et élaborez un ensemble de plans et de spécifications nécessaires à l'implémentation de la solution. L'approbation du projet intervient au cours de cette phase.

Il se compose des sections suivantes:

- "À propos de la conception du déploiement" à la page 70
- "Méthodologie de conception du déploiement" à la page 73
- "Estimation de la puissance de traitement" à la page 74
- "Estimation de la puissance de traitement requise pour les transactions sécurisées " à la page 80
- "Identification des stratégies de disponibilité" à la page 83
- "Identification des stratégies d'évolutivité" à la page 91
- "Identification des goulots d'étranglement des performances" à la page 93
- "Conception pour une utilisation optimale des ressources" à la page 96
- "Gestion des risques" à la page 97
- "Exemple d'architecture de déploiement" à la page 98

À propos de la conception du déploiement

La conception du déploiement se fonde sur le scénario de déploiement créé au cours des phases de conception logique et de spécification technique du cycle de vie de la solution. Un scénario de déploiement présente l'architecture logique associée aux exigences de qualité de service (QoS) pour cette solution. Pour créer une architecture de déploiement, vous mappez les composants identifiés dans l'architecture logique sur les serveurs physiques et les autres périphériques réseau. Les exigences de qualité de service permettent de configurer le matériel en vue d'obtenir des performances, une disponibilité et une évolutivité optimales.

La conception de l'architecture de déploiement est un processus itératif. Celui-ci consiste généralement à réexaminer les exigences de qualité de service et les conceptions préliminaires. Vous devez tenir compte des relations existant entre les différentes exigences de qualité de service ainsi que des problèmes liés aux coûts de possession et, après avoir évalué les avantages et les inconvénients, parvenir à une solution optimale permettant d'atteindre les objectifs d'exploitation du projet.

Approbation du projet

L'approbation du projet intervient au cours de la phase de conception du déploiement, généralement après la création de l'architecture de déploiement. À l'aide de l'architecture de déploiement et, éventuellement, des spécifications d'implémentation décrites ci-dessous, le coût réel du déploiement est estimé et soumis aux parties prenantes pour approbation. Une fois le projet approuvé, les contrats relatifs à l'exécution du déploiement sont signés et les ressources destinées à l'implémentation du projet sont acquises et allouées.

Résultats de la conception du déploiement

Lors de la phase de conception du déploiement, vous pouvez préparer les spécifications et les plans suivants :

- **Architecture de déploiement** : architecture de haut niveau qui décrit le mappage d'une architecture logique sur un environnement physique. Celui-ci comprend les nœuds de traitement dans un environnement intranet ou Internet, les processeurs, la mémoire, les périphériques de stockage et tout autre type de matériel et de périphérique réseau.
- **Spécifications d'implémentation** : spécifications détaillées servant à la planification du déploiement. Elles décrivent le matériel à acheter et la structure du réseau. Elles définissent également les services d'annuaire, y compris

l'arborescence d'informations d'annuaire, et les groupes et rôles définis pour l'accès à l'annuaire.

- **Plans d'implémentation** : groupe de plans couvrant divers aspects de l'implémentation d'une solution logicielle d'entreprise. Ces plans sont les suivants :
 - *Plan de migration* : décrit les stratégies et les processus de migration des données et de mise à niveau des logiciels de l'entreprise. Les données migrées doivent respecter les formats et les standards des nouvelles applications installées. Tous les logiciels de l'entreprise doivent avoir un niveau de version leur permettant d'interagir les uns avec les autres.
 - *Plan d'installation* : issu de l'architecture de déploiement, il spécifie les noms des serveurs matériels, les répertoires d'installation, la séquence d'installation, le type d'installation de chaque nœud ainsi que les informations de configuration requises pour installer et configurer un déploiement distribué.
 - *Plan de gestion des utilisateurs* : comprend des stratégies de migration pour les données des répertoires et des bases de données existants, des spécifications de structure des répertoires prenant en compte les mécanismes de réplication définis dans l'architecture de déploiement, ainsi que des procédures permettant d'alimenter les répertoires par un nouveau contenu.
 - *Plan de test* : décrit les procédures de test des logiciels déployés, y compris les plans permettant de développer des implémentations de prototypes et de pilotes, les tests destinés à mesurer la capacité du système à traiter la charge de travail, ainsi que les tests permettant de déterminer si une fonctionnalité se comporte comme prévu.
 - *Plan de déploiement* : présente les procédures et le planning utilisés pour faire passer l'implémentation de l'environnement de planification et de test à l'environnement de production. Cette opération se déroule généralement en plusieurs phases. La première phase peut par exemple consister à déployer le logiciel auprès d'un groupe réduit d'utilisateurs, puis d'augmenter progressivement le nombre de ces utilisateurs jusqu'à ce que le déploiement soit terminé. L'implémentation par phases peut également prévoir le déploiement progressif de logiciels spécifiques.
 - *Plan de reprise sur sinistre* : décrit les procédures de restauration du système après une panne. Ces procédures s'appliquent à tous les types de pannes, quelle que soit leur gravité.
 - *Plan de fonctionnement (manuel d'exploitation)* : manuel décrivant les procédures de contrôle, de maintenance, d'installation et de mise à niveau.
 - *Plan de formation* : contient les processus et les procédures de formation des opérateurs, des administrateurs et des utilisateurs finals sur le nouveau logiciel de l'entreprise.

Facteurs influant sur la conception du déploiement

Les décisions prises lors de la conception du déploiement sont influencées par un certain nombre de facteurs. Ces facteurs sont les suivants :

- **Architecture logique** : décrit en détail les services fonctionnels d'une solution possible et les relations existant entre les composants fournissant ces services. Servez-vous de l'architecture logique pour déterminer la meilleure manière de distribuer les services. Un scénario de déploiement présente l'architecture logique ainsi que les exigences en termes de qualité de service (décrites ci-dessous).
- **Exigences de qualité de service** : ces exigences précisent divers aspects du fonctionnement d'une solution. Utilisez-les pour développer des stratégies permettant d'atteindre les objectifs de qualité de service en termes de performances, de disponibilité, d'évolutivité, d'entretien, etc. Un scénario de déploiement présente l'architecture logique (décrite ci-dessus) ainsi que les exigences en termes de qualité de service.
- **Analyse d'utilisation** : l'analyse d'utilisation, développée au cours de la phase de spécification technique du cycle de vie de la solution, fournit des informations sur les types d'utilisation permettant d'évaluer la charge de travail d'un système déployé. Utilisez-la pour identifier les goulots d'étranglement des performances et élaborer des stratégies permettant de satisfaire les exigences de qualité de service.
- **Cas d'utilisation** : ces cas sont développés au cours de la phase de spécification technique du cycle de vie de la solution et répertorient les différentes interactions utilisateur identifiées au cours d'un déploiement. Bien qu'ils soient intégrés à l'analyse d'utilisation, vous devez examiner ces cas lors de l'évaluation de la conception d'un déploiement afin de vous assurer qu'ils sont bien pris en compte.
- **Accords de niveau de service** : un accord de niveau de service précise les performances minimales exigées et, au cas où ces exigences ne seraient pas satisfaites, le niveau et l'étendue du support client devant être fourni. Une conception de déploiement doit répondre aux exigences de performances définies dans un accord de niveau de service.
- **Coût total de possession** : lors de la conception du déploiement, vous analysez les solutions susceptibles de répondre aux exigences de qualité de service en termes de disponibilité, de performances, d'évolutivité, etc. Toutefois, pour chaque solution envisagée, vous devez également tenir compte du coût de cette solution et de son impact sur le coût total de possession. Veillez à prendre en compte les compromis qu'impliquent vos choix et à optimiser vos ressources de manière à répondre aux exigences de l'entreprise dans le respect des contraintes d'exploitation.
- **Objectifs d'exploitation** : ces objectifs sont définis au cours de la phase d'analyse d'exploitation de la solution. Ils tiennent compte des exigences de l'entreprise et des contraintes d'exploitation liées à leur réalisation. La capacité d'une conception de déploiement à répondre aux objectifs d'exploitation est la meilleure preuve de sa qualité.

Méthodologie de conception du déploiement

Au même titre que les autres aspects du déploiement, la conception ne relève pas uniquement de la science pure ; c'est pourquoi elle ne peut faire l'objet de procédures et de processus systématiques. L'expérience en matière de conception, une connaissance de l'architecture des systèmes, la compréhension du domaine et un effort de réflexion créative constituent autant de facteurs contribuant à la réussite d'un déploiement.

L'objectif principal de la conception du déploiement consiste à répondre aux exigences de performances tout en satisfaisant celles relatives à la qualité de service. Les stratégies adoptées doivent présenter des avantages malgré les compromis impliqués par vos décisions de conception afin d'optimiser la solution. La méthodologie utilisée comprend généralement les étapes suivantes :

- **Estimation de la puissance de traitement** : la conception du déploiement commence normalement par l'estimation du nombre de CPU requises pour chaque composant de l'architecture logique. Commencez par les cas d'utilisation correspondant à la charge la plus importante et poursuivez avec les autres cas. Considérez la charge incombant à tous les composants prenant en charge les cas d'utilisation et modifiez vos estimations en conséquence. Tenez également compte de vos expériences précédentes en matière de conception de systèmes d'entreprise.
- **Estimation de la puissance de traitement requise pour le transport sécurisé** : étudiez les cas nécessitant un transport sécurisé et modifiez le nombre de CPU requises en conséquence.
- **Réplication des services pour la disponibilité et l'évolutivité** : une fois la puissance de traitement estimée, modifiez la conception en fonction des exigences de qualité de service en termes de disponibilité et d'évolutivité. Envisagez de mettre en œuvre des solutions d'équilibrage de charge pour résoudre les problèmes de disponibilité et de basculement.

Au cours de l'analyse, tenez compte des implications que présentent vos décisions de conception. Par exemple, demandez-vous quel impact les stratégies de disponibilité et d'évolutivité peuvent avoir sur la maintenance du système. Renseignez-vous également sur les autres coûts induits par ces stratégies.
- **Identification des goulots d'étranglement** : dans les phases suivantes de votre analyse, examinez la conception du déploiement afin d'identifier les goulots d'étranglement susceptibles de nuire aux performances de transmission des données et apportez les modifications nécessaires.
- **Optimisation des ressources** : étudiez la conception du déploiement sous l'angle de la gestion des ressources et envisagez des solutions permettant de réduire les coûts tout en répondant aux exigences formulées.

- **Gestion des risques** : réexaminez vos analyses d'exploitation et vos analyses techniques et modifiez votre conception en fonction d'éventuels événements ou situations qui n'auraient pas été envisagés lors des phases précédentes de la planification.

Estimation de la puissance de traitement

Cette section décrit un processus permettant d'estimer le nombre de processeurs et la quantité de mémoire nécessaires pour prendre en charge les services d'une conception de déploiement. Elle présente un exemple de processus appliqué à un déploiement dans le secteur des communications.

L'estimation de la puissance de traitement des CPU est un processus itératif prenant en compte les éléments suivants :

- les composants logiques et leurs interactions (définies par les dépendances entre composants dans l'architecture logique) ;
- l'analyse des cas d'utilisation identifiés ;
- les exigences en termes de qualité de service ;
- l'expérience acquise en matière de conception de déploiements et avec Java Enterprise System ;
- les conseils des services professionnels de Sun, familiarisés avec la conception et l'implémentation de divers types de scénarios de déploiement.

Le processus d'estimation s'articule autour des étapes décrites ci-dessous. Bien qu'il ne soit pas essentiel, l'ordre de ces étapes fournit un exemple de méthode à suivre lors de la prise en compte des facteurs susceptibles d'influencer le résultat final.

1. Définissez une estimation du nombre de CPU requises pour les composants identifiés en tant que points d'entrée utilisateur dans le système.

Vous devez décider si les CPU doivent supporter une charge complète ou partielle. Une charge complète augmente la capacité du système. L'augmentation de la capacité vous expose néanmoins à une hausse des coûts de maintenance et à des temps d'indisponibilité dus à l'ajout de CPU. Dans certains cas, vous pouvez décider d'ajouter des machines pour répondre à l'augmentation des exigences en matière de performances.

Une charge partielle permet de faire face aux exigences de performances supplémentaires sans hausse immédiate des coûts de maintenance. Toutefois, un système sous-utilisé entraîne une augmentation des frais directs.

2. Modifiez l'estimation du nombre de CPU requises en fonction des interactions entre composants.

Examinez ces interactions dans l'architecture logique afin de déterminer la charge supplémentaire entraînée par les dépendances entre composants.

3. Dans l'analyse d'utilisation, examinez les cas d'utilisation afin d'identifier les charges de pointe, puis effectuez les ajustements nécessaires pour les composants qui gèrent ces charges.

Commencez par les cas pour lesquels la charge est la plus importante, puis passez en revue tous les autres cas pour prendre en compte l'intégralité des scénarios d'utilisation prévus.

4. Modifiez le nombre de CPU estimé en fonction des exigences de sécurité, de disponibilité et d'évolutivité.

Les estimations établies constituent un point de départ à partir duquel déterminer la puissance de traitement réelle dont vous avez besoin. Elles vous permettront également de créer des prototypes de déploiement, que vous soumettrez à des tests rigoureux en fonction des cas d'utilisation prévus. Seul un test itératif vous permettra de déterminer les besoins réels d'une conception de déploiement en termes de puissance de traitement.

Exemple d'estimation de la puissance de traitement

Cette section illustre une méthode d'estimation de la puissance de traitement requise pour un exemple de déploiement. Cet exemple est fondé sur l'architecture logique d'une solution de communications basées sur les identités pour une entreprise de taille moyenne d'environ 1 000 à 5 000 salariés, tel que décrit dans la section ["Exemple de communications basées sur les identités"](#) à la page 63.

Les estimations mentionnées pour les CPU et la mémoire sont arbitraires et fournies à titre d'exemple uniquement. Elles sont fondées sur les données arbitraires sur lesquelles s'appuie l'exemple. Seule une analyse exhaustive de divers facteurs permet d'établir les besoins en termes de puissance de traitement. Cette analyse porte notamment sur les éléments suivants :

- cas d'utilisation détaillés et analyse d'utilisation fondée sur une analyse d'exploitation exhaustive ;
- exigences de qualité de service déterminée par l'analyse des exigences de l'entreprise ;
- coûts particuliers et spécifications du matériel réseau et de traitement ;
- expérience acquise lors de déploiements similaires.



Attention – Les informations présentées dans les exemples ci-après ne constituent pas des conseils d'implémentation à proprement parler. Elles sont uniquement destinées à illustrer un processus que vous serez peut-être amené à utiliser lors de la conception d'un système.

Estimation du nombre de CPU requises pour les points d'entrée utilisateur

Commencez par évaluer le nombre de CPU nécessaires pour gérer la charge incombant à chaque composant représentant un point d'entrée utilisateur. Le schéma ci-dessous illustre l'architecture logique d'un scénario de communications basées sur les identités décrit dans la section "Exemple de communications basées sur les identités" à la page 63.

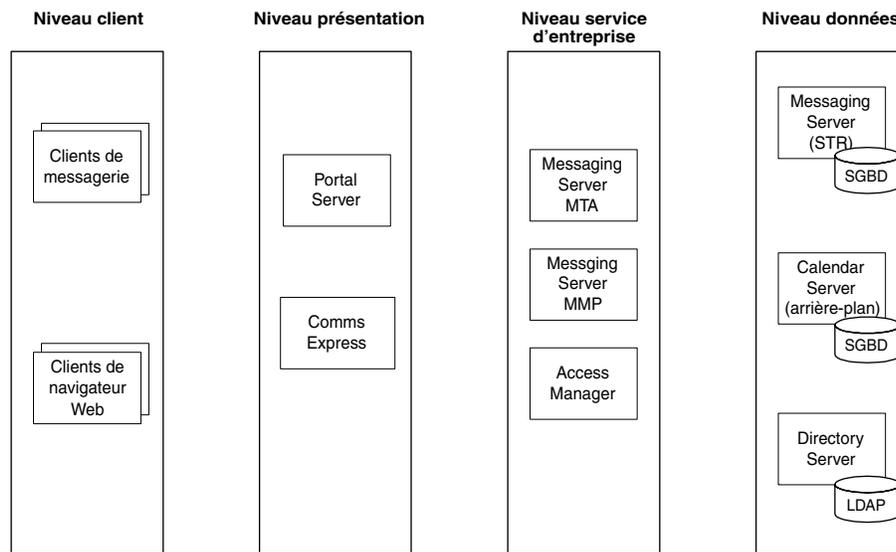


FIGURE 5-1 Architecture logique d'un scénario de communications basées sur les identités

Le tableau ci-dessous répertorie les composants du niveau présentation de l'architecture logique qui fournissent une interface directe aux utilisateurs finals du déploiement. Il présente une estimation de base du nombre de CPU requises, obtenue à partir de l'analyse des exigences techniques, des cas d'utilisation, de l'analyse d'utilisation spécifique et de l'expérience acquise grâce à des déploiements similaires.

TABLEAU 5-1 Estimation du nombre de CPU pour les composants contenant des points d'entrée utilisateur

Composant	Nombre de CPU	Description
Portal Server	4	Composant constituant un point d'entrée utilisateur.
Communications Express	2	Achemine les données vers les canaux de messagerie et de calendrier de Portal Server.

Inclusion du nombre de CPU estimé pour les services dépendants

Les composants fournissant des points d'entrée utilisateur doivent être secondés par d'autres composants Java Enterprise System. Au cours de la définition des exigences de performances, incluez les estimations de performances requises pour les composants de support. Les types d'interactions entre composants doivent être décrits en détail lors de la conception de l'architecture logique, comme le montrent les exemples d'architecture logique présentés à la section "Exemple d'architecture logique" à la page 58.

TABLEAU 5-2 Estimation du nombre de CPU pour les composants de support

Composant	Nombre de CPU	Description
Messaging Server MTA (entrant)	1	Achemine les messages entrants depuis Communications Express et les clients de messagerie.
Messaging Server MTA (sortant)	1	Achemine les messages sortants vers les destinataires.
Messaging Server MMP	1	Accède à la mémoire des messages de Messaging Server pour les clients de messagerie.
Messaging Server STR (Message Store)	1	Extrait et stocke les messages.
Access Manager	2	Fournit des services d'authentification et d'autorisation.
Calendar Server (arrière-plan)	2	Extrait et stocke les données de calendrier pour Communications Express, composant frontal de Calendar Server.
Directory Server	2	Fournit les services d'annuaire LDAP.
Web Server	0	Fournit la prise en charge de conteneur Web pour Portal Server et Access Manager. (Aucun cycle de CPU supplémentaire n'est nécessaire.)

Étude des cas d'utilisation pour les charges de pointe

Revenez aux cas d'utilisation et à l'analyse d'utilisation pour identifier les situations de charge de pointe et modifiez vos estimations en conséquence.

Toujours dans le cadre de notre exemple, supposons que vous relevez les conditions de charge de pointe suivantes :

- pic de trafic initial lorsque les utilisateurs se connectent simultanément ;

- échanges de messages au cours de certaines périodes.

Pour prendre en compte ces conditions, vous devez effectuer des ajustements pour les composants fournissant ces services. Le tableau ci-dessous décrit les ajustements à effectuer.

TABLEAU 5-3 Ajustement de l'estimation du nombre de CPU pour les charges de pointe

Composant	Nombre de CPU (ajusté)	Description
Messaging Server MTA (entrant)	2	Ajoutez une CPU pour les messages entrants supplémentaires
Messaging Server MTA (sortant)	2	Ajoutez une CPU pour les messages sortants supplémentaires
Messaging Server MMP	2	Ajoutez une CPU pour la charge supplémentaire
Messaging Server STR (Message Store)	2	Ajoutez une CPU pour la charge supplémentaire
Directory Server	3	Ajoutez une CPU pour les recherches LDAP supplémentaires

Modification des estimations pour les autres conditions de charge

Poursuivez votre estimation du nombre de CPU en tenant compte des autres exigences de qualité de service susceptibles d'avoir un impact sur la charge :

- **Sécurité** : au cours de la phase de spécification technique, déterminez si le transport sécurisé des données peut affecter les exigences de charge et modifiez vos estimations en conséquence. La section ["Estimation de la puissance de traitement requise pour les transactions sécurisées "](#) à la page 80 explique comment procéder à ces modifications.
- **Réplication des services** : ajustez vos estimations en tenant compte de la réplication des services à des fins de disponibilité, d'équilibrage de charge et d'évolutivité. La section suivante, ["Identification des stratégies de disponibilité"](#) à la page 83, décrit le concept de dimensionnement dans le cadre des solutions de disponibilité. La section ["Identification des stratégies d'évolutivité"](#) à la page 91 décrit les solutions permettant un accès efficace aux services d'annuaire.
- **Capacité latente et évolutivité** : modifiez le nombre de CPU estimé en fonction de la capacité latente requise pour faire face aux charges importantes et imprévues. Examinez les étapes de mise à l'échelle et l'augmentation de la charge prévues et assurez-vous que toutes les dates planifiées pour la mise à l'échelle horizontale ou verticale du système pourront être respectées.

Mise à jour du nombre de CPU estimé

Il est généralement préférable d'avoir un nombre pair de CPU. Cela permet de diviser équitablement le nombre de CPU entre deux serveurs physiques et d'ajouter un petit facteur pour la capacité latente. Toutefois, lorsque vous définissez le nombre de CPU, tenez compte de vos besoins spécifiques en matière de réplication des services.

Comptez normalement deux gigaoctets de mémoire pour chaque CPU. La quantité réelle de mémoire requise dépend de vos besoins spécifiques. Elle peut être déterminée en phase de test.

Le tableau ci-dessous indique les estimations finales pour l'exemple de communications basées sur les identités. Ces estimations ne tiennent pas compte de la puissance de traitement éventuellement ajoutée à des fins de sécurité et de disponibilité. Ces valeurs seront ajoutées dans les sections qui suivent.

TABLEAU 5-4 Ajustement de l'estimation du nombre de CPU pour les composants de support

Composant	Nombre de CPU	Mémoire
Portal Server	4	8 Go
Communications Express	2	4 Go
Messaging Server (MTA, entrant)	2	4 Go
Messaging Server (MTA, sortant)	2	4 Go
Messaging Server (MMP)	2	4 Go
Messaging Server (Message Store)	2	4 Go
Access Manager	2	4 Go
Calendar Server	2	4 Go
Directory Server	4	8 Go (valeur arrondie à partir de 3 CPU pour 6 Go de mémoire)
Web Server	0	0

Estimation de la puissance de traitement requise pour les transactions sécurisées

Le transport sécurisé de données implique le traitement des transactions par le biais d'un protocole de transport sécurisé tel SSL (Secure Sockets Layer) ou TLS (Transport Layer Security). Ces transactions exigent généralement une puissance de traitement supplémentaire pour établir une session sécurisée (handshake), puis pour chiffrer et déchiffrer les données transportées. Selon l'algorithme de chiffrement utilisé (40 ou 128 bits, par exemple), cette puissance supplémentaire peut être significative.

Pour que les transactions sécurisées puissent s'exécuter au même niveau que les transactions non sécurisées, vous devez prévoir une puissance de traitement supplémentaire. Selon leur nature et les services Sun Java™ Enterprise System qui les gèrent, les transactions sécurisées peuvent exiger une puissance de traitement jusqu'à quatre fois supérieure à celle des transactions non sécurisées.

Lors de l'évaluation de la puissance de traitement destinée à gérer les transactions sécurisées, analysez les cas d'utilisation afin de déterminer le pourcentage de transactions nécessitant un transport sécurisé. Si les exigences de performances des transactions sécurisées sont identiques à celles des transactions non sécurisées, modifiez l'estimation du nombre de CPU en tenant compte de la puissance de traitement supplémentaire requise par les transactions sécurisées.

Dans certains scénarios d'utilisation, le transport sécurisé est requis uniquement pour l'authentification. Une fois l'utilisateur authentifié sur le système, aucune mesure de sécurité supplémentaire n'est appliquée au transport des données. Dans d'autres scénarios, le transport sécurisé est exigé pour toutes les transactions.

Par exemple, lors de la consultation d'un catalogue de produits sur un site d'e-commerce en ligne, il n'est pas nécessaire de sécuriser les transactions tant que le client n'a pas choisi tous ses articles et qu'il n'est pas prêt à effectuer le paiement. Toutefois, dans certains scénarios de déploiement appliqués à des banques ou à des agences immobilières, par exemple, la plupart des transactions doivent être sécurisées et les mêmes performances sont attendues de toutes les transactions, qu'elles soient ou non sécurisées.

Estimation du nombre de CPU pour les transactions sécurisées

Cette section utilise le même exemple de déploiement pour illustrer le mode de calcul du nombre de CPU nécessaires pour un cas d'utilisation hypothétique mettant en œuvre à la fois des transactions sécurisées et non sécurisées.

Pour estimer le nombre de CPU requises pour les transactions sécurisées, effectuez les calculs suivants :

1. Partez d'une estimation du nombre de CPU requises (voir la section précédente, "Exemple d'estimation de la puissance de traitement" à la page 75).
2. Calculez le pourcentage de transactions exigeant un transport sécurisé, puis le nombre de CPU nécessaires pour ces transactions.
3. Comptez un nombre de CPU inférieur pour les transactions non sécurisées.
4. Additionnez les valeurs obtenues pour parvenir à une estimation du nombre total de CPU nécessaires.
5. Arrondissez cette estimation à une valeur paire.

La section "Estimation du nombre de CPU pour les transactions sécurisées" à la page 80 montre un exemple de calcul fondé sur des cas d'utilisation et une analyse d'utilisation pour Portal Server et tenant compte des hypothèses suivantes :

- Toutes les connexions exigent une authentification sécurisée.
- Les connexions représentent 10% de la charge totale de Portal Server.
- Les exigences de performances relatives aux transactions sécurisées et non sécurisées sont identiques.

Le nombre de CPU requises sera multiplié par quatre afin de fournir la puissance de traitement supplémentaire pour les transactions sécurisées. Comme les autres valeurs citées dans cet exemple, ce facteur est arbitraire et fourni à titre d'exemple uniquement.

TABLEAU 5-5 Modification du nombre de CPU estimé pour les transactions sécurisées

Étape	Description	Calcul	Résultat
1	Partez d'une estimation de base pour toutes les transactions Portal Server.	Cette estimation, issue de la section "Étude des cas d'utilisation pour les charges de pointe" à la page 77 est de 4 CPU.	----
2	Calculez l'estimation du nombre de CPU supplémentaires pour les transactions sécurisées. Supposez que celles-ci exigent une puissance de traitement quatre fois supérieure à celle des transactions non sécurisées.	10% des transactions (estimation de base) doivent faire l'objet d'un transport sécurisé : $0.10 \times 4 \text{ CPUs} = 0.4 \text{ CPUs}$ Augmentez la puissance de traitement pour les transactions sécurisées selon un facteur de 4 : $4 \times 0.4 = 1.6 \text{ CPUs}$	1,6 CPU
3	Comptez un nombre de CPU inférieur pour les transactions non sécurisées.	90% des transactions (estimation de base) ne sont pas sécurisées : $0.9 \times 4 \text{ CPUs} = 3.6 \text{ CPUs}$	3,6 CPU

TABLEAU 5-5 Modification du nombre de CPU estimé pour les transactions sécurisées
(Suite)

Étape	Description	Calcul	Résultat
4	Calculez le nombre total de CPU estimé pour les transactions sécurisées et non sécurisées.	Transactions sécurisées + non sécurisées = total : $1.6 \text{ CPU}_s + 3.6 \text{ CPU}_s = 5.2 \text{ CPU}_s$	5,2 CPU
5	Arrondissez le résultat à une valeur paire.	$5.2 \text{ CPU}_s \implies 6 \text{ CPU}_s$	6 CPU

Sur la base des calculs effectués pour les transactions sécurisées dans cet exemple, vous devez modifier le nombre total de CPU estimé à la section [“Estimation du nombre de CPU pour les transactions sécurisées”](#) à la page 80 en ajoutant deux CPU et quatre gigaoctets de mémoire afin de parvenir au total ci-dessous pour Portal Server :

Composant	Nombre de CPU	Mémoire
Portal Server	6	12 Go

Matériel dédié au traitement des transactions SSL

Du matériel spécialisé, tel des cartes d'accélération SSL et d'autres équipements, est disponible pour fournir la puissance de traitement nécessaire à l'établissement de sessions sécurisées et au chiffrement/déchiffrement des données. Lorsque ce type de matériel est utilisé, une partie de la puissance de traitement est consacrée à certains calculs SSL, notamment l'opération de « handshake » permettant d'établir la session sécurisée.

L'utilisation de ce type de matériel peut être un avantage pour votre architecture de déploiement. Toutefois, en raison de son caractère spécialisé, il est préférable d'estimer les exigences de performances des transactions sécurisées d'abord en termes de puissance de traitement, puis d'envisager les avantages liés à l'utilisation de ce matériel pour gérer la charge supplémentaire.

Avant d'utiliser ce type de matériel, vous devez déterminer s'il est pris en charge par les cas d'utilisation (ceux qui nécessitent un grand nombre d'opérations de handshake SSL, par exemple) et tenir compte du surcroît de complexité qu'il engendre en termes de conception. Cette complexité se retrouve dans les opérations d'installation, de configuration, de test et d'administration du matériel.

Identification des stratégies de disponibilité

Lors de l'établissement d'une stratégie relative aux exigences de disponibilité, examinez les interactions entre composants et l'analyse d'utilisation pour identifier les solutions de disponibilité à envisager. Effectuez cette analyse composant par composant pour trouver la solution répondant le mieux aux exigences de disponibilité et de basculement.

Vous pouvez par exemple vous poser les questions suivantes pour mettre en place votre stratégie de disponibilité :

- Quel est le degré de disponibilité spécifié (quatre ou cinq neuf, par exemple) ?
- Quelles sont les spécifications de performances en cas de basculement (50 %, par exemple) ?
- L'analyse d'utilisation identifie-t-elle des périodes d'utilisation de pointe ?
- Quelles sont les considérations géographiques à prendre en compte ?

La stratégie de disponibilité choisie doit également tenir compte des exigences en termes d'entretien (voir section "[Conception pour une utilisation optimale des ressources](#)" à la page 96). Évitez les solutions complexes exigeant des efforts de maintenance et d'administration importants.

Stratégies de disponibilité

Les stratégies de disponibilité possibles pour les déploiements Java Enterprise System sont les suivantes :

- *Équilibrage de charge* : utilisez des composants matériels et logiciels redondants pour répartir la charge de traitement. Un équilibreur de charge dirige une demande de service vers une des instances symétriques de ce service. Si une instance échoue, les autres peuvent assumer sa charge de travail.
- *Basculement* : implique la gestion de matériels et de logiciels redondants afin que l'accès aux services et que la sécurité des données essentielles ne soient pas compromis en cas de problème lié à un composant.

Le logiciel Sun Cluster propose une solution de basculement pour les données essentielles gérées par des composants d'arrière-plan tels la mémoire des messages pour Messaging Server et les données de calendrier pour Calendar Server.

- *Réplication des services* : cette fonction fournit plusieurs sources d'accès aux mêmes données. Directory Server propose de nombreuses stratégies de réplication et de synchronisation pour l'accès aux annuaires LDAP.

Les sections ci-après décrivent des exemples de solutions de disponibilité fournissant divers niveaux d'équilibrage de charge, de basculement et de réplication des services.

Système à serveur unique

Toutes les ressources de traitement d'un service sont placées sur un seul serveur. Si ce dernier tombe en panne, le service ne peut plus fonctionner.

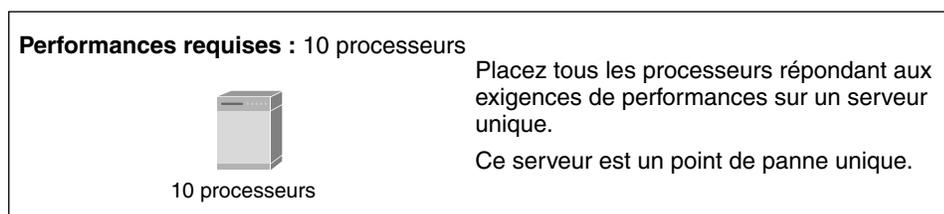


FIGURE 5-2 Système à serveur unique

Sun fournit des serveurs haut de gamme permettant de bénéficier des avantages suivants :

- remplacement et reconfiguration des composants matériels en cours d'exécution du système ;
- possibilité d'exécuter plusieurs applications dans des domaines sécurisés sur le serveur ;
- possibilité de mettre à niveau la capacité, les performances et la configuration des E/S sans redémarrer le système.

Un serveur haut de gamme coûte généralement plus cher qu'un système multiserveur comparable. Toutefois, il se révèle plus économique en termes d'administration, de contrôle et d'hébergement. Les systèmes multiserveurs offrent une plus grande souplesse en termes d'équilibrage de charge, de basculement et de suppression de points de panne uniques.

Systèmes à redondance horizontale

Les serveurs redondants parallèles offrant des fonctions d'équilibrage de charge et de basculement, ils permettent d'améliorer la disponibilité de plusieurs façons. Le schéma ci-dessous montre deux serveurs répliqués constituant un système à basculement N+1. Dans ce type de système, un serveur supplémentaire est destiné à assumer 100 % de la charge en cas de panne de l'autre serveur.

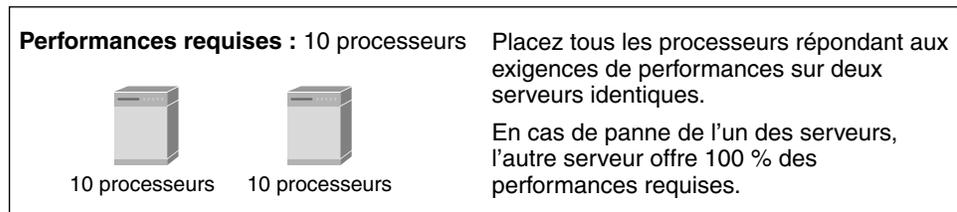


FIGURE 5-3 Système à basculement N+1 avec deux serveurs

La puissance de traitement de chaque serveur dans des “[Systèmes à redondance horizontale](#)” à la page 84 (voir ci-dessus) est identique. Un serveur peut à lui seul satisfaire les exigences de performances. L'autre serveur offre exactement les mêmes performances lorsqu'il est utilisé en tant que serveur de rechange.

Dans ce type de conception, 100% des performances sont assurées en cas de basculement. En revanche, les investissements matériels élevés sans amélioration des performances (puisque l'un des serveurs est utilisé en cas de basculement uniquement) représentent un inconvénient majeur.

Le schéma suivant illustre un système mettant en œuvre l'équilibrage de charge et le basculement, et répartissant les performances entre deux serveurs.

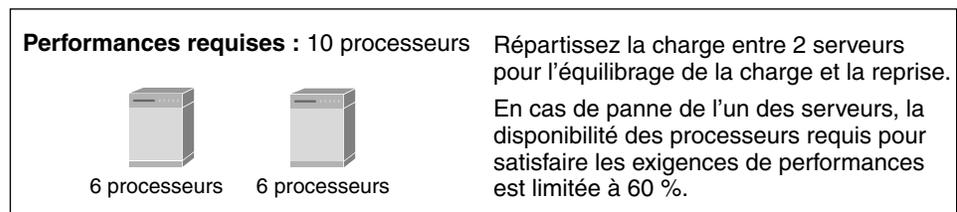


FIGURE 5-4 Équilibrage de charge et basculement entre deux serveurs

Dans le système représenté à la section “[Systèmes à redondance horizontale](#)” à la page 84, si un serveur tombe en panne, tous les services restent disponibles, mais pas à 100 %. En effet, le second serveur comporte six CPU qui permettent de prendre en charge 60 % de la charge totale (10 CPU).

Cette conception offre l'avantage d'autoriser une capacité latente (deux CPU) lorsque les deux serveurs sont disponibles.

Le schéma ci-dessous illustre une répartition des performances et de l'équilibrage de charge entre plusieurs serveurs.

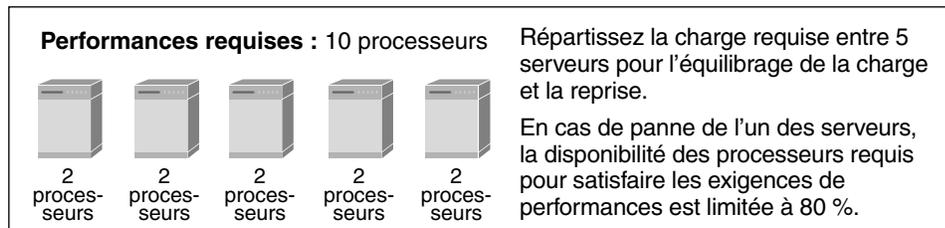


FIGURE 5-5 Répartition de la charge entre n serveurs

La conception illustrée à la section “[Systèmes à redondance horizontale](#)” à la page 84 comportant cinq serveurs, si l’un d’eux tombe en panne, les autres fournissent un total de huit CPU, soit 80 % des exigences de performances (10 CPU). Si vous ajoutez un serveur doté de deux CPU, vous obtenez une conception N+1. En cas de panne de l’un des serveurs, 100 % des performances sont maintenues grâce aux autres serveurs.

Cette solution offre les avantages suivants :

- performance accrue en cas de panne d’un serveur ;
- disponibilité même en cas de panne de plusieurs serveurs ;
- possibilité de mettre les serveurs hors service pour la maintenance et les mises à niveau ;
- dépenses réduites, car plusieurs serveurs d’entrée de gamme coûtent moins cher qu’un seul serveur haut de gamme.

Toutefois, plus le nombre de serveurs augmente, plus les coûts d’administration et de maintenance sont élevés, sans compter les frais d’hébergement des serveurs dans un centre de données. De plus, il arrive un moment où l’ajout de serveurs finit par entraîner plus d’inconvénients que d’avantages.

Logiciel Sun Cluster

Dans les situations exigeant un degré élevé de disponibilité (quatre ou cinq neuf), vous pouvez envisager d’intégrer le logiciel Sun Cluster dans votre conception de déploiement. Un cluster associe des serveurs redondants à des ressources de stockage et d’autres ressources réseau. Les serveurs d’un cluster sont en communication permanente les uns avec les autres. En cas de mise hors ligne d’un serveur, les autres dispositifs du cluster l’isolent et font basculer les applications ou les données du nœud en panne vers un autre nœud. Le basculement s’effectue rapidement sans que les utilisateurs du système aient à subir une longue interruption des services.

Sun Cluster exige du matériel supplémentaire spécialisé et des connaissances particulières en matière de configuration, d’administration et de maintenance.

Exemples de conceptions de disponibilité

Cette section présente deux exemples de stratégies de disponibilité appliqués à une solution de communications basées sur les identités pour une entreprise de taille moyenne de 1 000 à 5 000 salariés, tel que décrit dans la section [“Exemple de communications basées sur les identités”](#) à la page 63. La première stratégie illustre l'équilibrage de charge pour Messaging Server. La seconde illustre une solution de basculement faisant appel au logiciel Sun Cluster.

Exemple d'équilibrage de charge pour Messaging Server

Le tableau ci-dessous présente le nombre de CPU requises estimé pour chaque composant logique de Messaging Server dans l'architecture logique. Il reprend l'estimation finale calculée dans la section [“Mise à jour du nombre de CPU estimé”](#) à la page 79 .

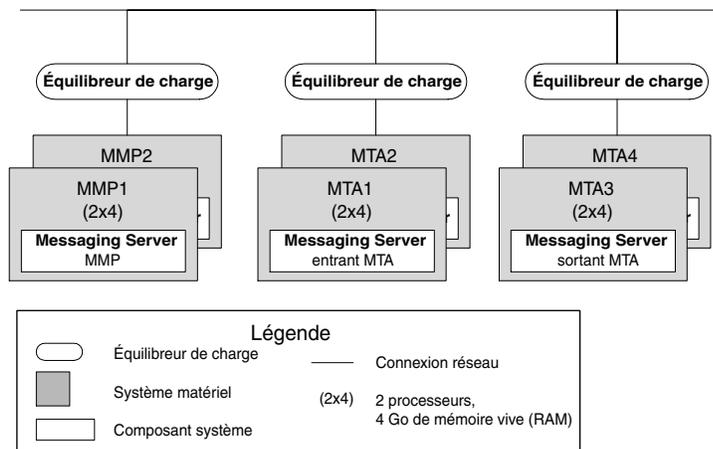
TABLEAU 5-6 Ajustement de l'estimation du nombre de CPU pour les composants de support

Composant	Nombre de CPU	Mémoire
Messaging Server (MTA, entrant)	2	4 Go
Messaging Server (MTA, sortant)	2	4 Go
Messaging Server (MMP)	2	4 Go
Messaging Server (Message Store)	2	4 Go

Pour cet exemple, supposons qu'au cours de la phase de spécification technique, les exigences de qualité de service suivantes aient été formulées :

- **Disponibilité** : la disponibilité globale du système doit être de 99,99% (hors périodes d'indisponibilité planifiées). Une panne d'un seul système ne doit pas entraîner d'interruption des services.
- **Évolutivité** : aucun serveur ne doit être utilisé à plus de 80% au cours des périodes de charge de pointe et le système doit pouvoir assumer une croissance à long terme de 10% par an.

Pour répondre aux exigences de disponibilité, prévoyez deux instances de chaque composant de Messaging Server, une sur chaque serveur. Si un serveur ou un composant ne fonctionne plus, l'autre le remplace. Le schéma ci-dessous présente un diagramme du réseau correspondant à cette stratégie de disponibilité.



Sur le schéma ci-dessus, le nombre de CPU a doublé par rapport à l'estimation initiale. Cette augmentation est intervenue pour les raisons suivantes :

- En cas de panne d'un serveur, l'autre fournit la puissance de traitement nécessaire à la gestion de la charge.
- La puissance de traitement supplémentaire fournit la marge de sécurité nécessaire pour répondre à l'exigence selon laquelle aucun serveur ne doit être utilisé à plus de 80% au cours des périodes de charge de pointe.
- En ce qui concerne l'exigence d'évolutivité impliquant une croissance annuelle de la charge de 10%, la puissance de traitement supplémentaire permet d'ajouter la capacité latente requise.

Exemple de basculement à l'aide du logiciel Sun Cluster

Le schéma ci-dessous montre un exemple de stratégie de basculement pour le composant d'arrière-plan Calendar Server et la mémoire de messages de Messaging Server. Le composant d'arrière-plan Calendar Server et la mémoire de messages sont répliqués sur des serveurs matériels différents et configurés pour le basculement à l'aide du logiciel Sun Cluster. Le nombre de CPU et la mémoire correspondante sont répliqués sur chaque serveur dans Sun Cluster.

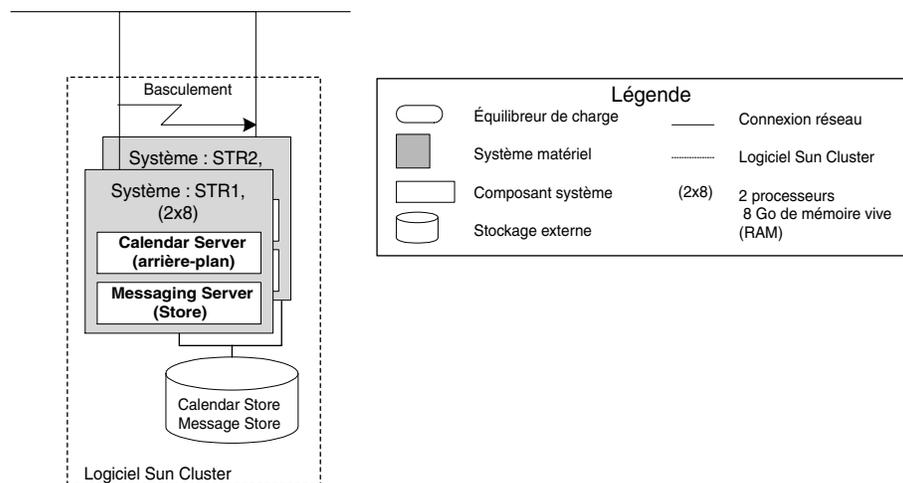


FIGURE 5-6 Conception avec basculement utilisant le logiciel Sun Cluster

Exemple de réplcation des services d'annuaire

Les services d'annuaire peuvent être répliqués de sorte que les transactions soient réparties sur différents serveurs, ce qui accroît la disponibilité. Directory Server fournit diverses stratégies de réplcation des services, dont les suivants :

- **Bases de données multiples** : les différentes parties d'une arborescence d'annuaire sont stockées dans des bases de données distinctes.
- **Chaînage et références** : relie les données distribuées dans une même arborescence d'annuaire.
- **Réplcation monomaître** : fournit une source centrale pour la base de données maître, qui est ensuite distribuée aux répliques consommateur.
- **Réplcation multimaître** : répartit la base de données maître entre plusieurs serveurs. Chacun d'eux répartit ensuite sa base de données entre les répliques consommateur.

Les stratégies de disponibilité pour Directory Server sont un sujet complexe que le présent manuel ne traite pas. Les sections "Réplcation monomaître" à la page 89 et "Réplcation multimaître" à la page 90 fournissent une présentation générale des stratégies de réplcation de base. Pour plus d'informations, reportez-vous au chapitre 12, "Designing a Highly Available Deployment," dans *Sun Java System Directory Server 5 2005Q1 Deployment Planning Guide*.

Réplcation monomaître

Le schéma ci-dessous représente une stratégie de réplcation monomaître illustrant les concepts de réplcation de base.

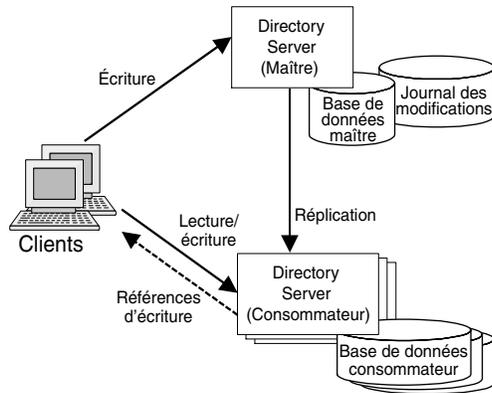


FIGURE 5-7 Exemple de réplification monomaître

Dans le cadre de la réplification monomaître, une instance de Directory Server gère la base de données d’annuaire maître et consigne toutes les modifications. La base de données maître est répliquée sur un certain nombre de bases de données consommateur. Les instances consommateur de Directory Server sont optimisées pour les opérations de lecture et de recherche. Toute opération d’écriture reçue par un consommateur est renvoyée vers le maître, et celui-ci met à jour les bases de données consommateur de manière régulière.

Les avantages de la réplification monomaître sont les suivants :

- instance unique de Directory Server optimisée pour les opérations de lecture et d’écriture dans la base de données ;
- nombre quelconque d’instances consommateur de Directory Server optimisées pour les opérations de lecture et de recherche ;
- évolutivité horizontale pour les instances consommateur de Directory Server.

Réplification multimaître

Le schéma ci-dessous illustre une stratégie de réplification multimaître permettant une distribution globale des accès à l’annuaire.

Avec la réplification multimaître, une ou plusieurs instances de Directory Server gèrent la base de données d’annuaire maître. Chacun de ces maîtres dispose d’un contrat de réplification décrivant les procédures de synchronisation des bases de données maître. Chaque maître est répliqué sur un nombre quelconque de bases de données consommateur. Comme dans le cadre de la réplification monomaître, les instances consommateur de Directory Server sont optimisées pour l’accès en lecture et en recherche. Toute opération d’écriture reçue par un consommateur est renvoyée vers le maître, et celui-ci met à jour les bases de données consommateur de manière régulière.

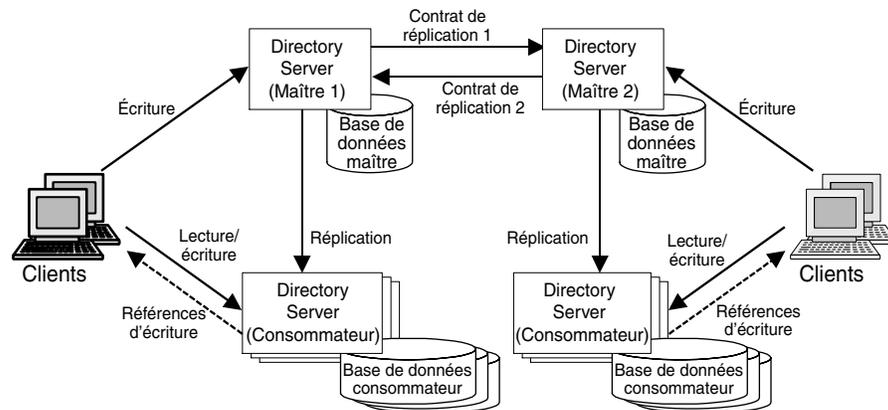


FIGURE 5-8 Exemple de réplication multimaître

La stratégie de réplication multimaître présente les mêmes avantages que la réplication monomaître, auxquels vient s'ajouter une disponibilité permettant l'équilibrage de charge lors des mises à jour du maître. Vous pouvez également implémenter une stratégie de disponibilité fournissant un contrôle local des opérations d'annuaire, importante pour les entreprises utilisant des centres de données distribués.

Identification des stratégies d'évolutivité

L'évolutivité se définit comme la possibilité d'augmenter la capacité du système, généralement via l'ajout de ressources système, sans pour autant modifier l'architecture de déploiement. Au cours de l'analyse des exigences, vous planifiez la croissance prévue du système en fonction des exigences de l'entreprise et des analyses d'utilisation correspondantes. Ces estimations du nombre d'utilisateurs et de la capacité du système à répondre à leurs besoins peuvent être très différentes des chiffres réels observés sur le système déployé. Votre conception doit fournir la souplesse nécessaire pour prendre en compte les différences par rapport aux estimations.

Une conception évolutive prévoit une capacité latente suffisante pour gérer les augmentations de charge de travail, jusqu'à ce que des ressources supplémentaires soient ajoutées au système. Elle doit pouvoir être adaptée sans être entièrement repensée.

Capacité latente

La capacité latente est un aspect de l'évolutivité consistant à ajouter des ressources de performances et de disponibilité au système de sorte que celui-ci soit en mesure de faire face aux charges de pointe. Dans un système déployé, vous pouvez également contrôler son utilisation afin de déterminer le moment où des ressources doivent être ajoutées au système. La capacité latente est un facteur de sécurité de votre conception.

L'analyse des cas d'utilisation peut permettre d'identifier les scénarios susceptibles d'entraîner des charges de pointe inhabituelles. Utilisez cette analyse et un facteur représentant la croissance imprévue du système pour définir la capacité latente et améliorer la sécurité du système.

La conception du système doit permettre de gérer la capacité prévue pendant un délai raisonnable, c'est-à-dire pendant les 6 à 12 premiers mois d'utilisation. Les cycles de maintenance peuvent être utilisés pour ajouter des ressources ou augmenter la capacité en cas de besoin. Idéalement, vous devriez pouvoir planifier des mises à niveau régulières du système, mais il est généralement difficile de prévoir les augmentations de capacité nécessaires. Pour savoir quand le système doit être mis à niveau, contrôlez attentivement vos ressources et fondez-vous sur les prévisions d'exploitation.

Si vous envisagez d'implémenter votre solution par phases incrémentielles, vous pouvez planifier les augmentations de capacité du système en même temps que les modifications prévues lors de chaque phase incrémentielle.

Exemple d'évolutivité

L'exemple de cette section illustre une mise à l'échelle horizontale et verticale pour une solution implémentant Messaging Server. La mise à l'échelle verticale consiste à ajouter des CPU à un serveur afin de gérer les augmentations de charge. La mise à l'échelle horizontale consiste à ajouter des serveurs supplémentaires afin de répartir la charge.

Cet exemple s'appuie sur une hypothèse de 50 000 utilisateurs et de deux instances de mémoire de messages réparties pour l'équilibrage de charge. Chaque serveur est équipé de deux CPU, pour un total de quatre. Le schéma ci-dessous montre comment ce système peut être mis à l'échelle pour gérer des charges de 250 000 et de 2 000 000 d'utilisateurs.

Remarque – L'"Exemple d'évolutivité" à la page 92 met en évidence les différences existant entre une mise à l'échelle verticale et une mise à l'échelle horizontale. Ce schéma ne prend pas en compte les autres facteurs à considérer lors de la mise à l'échelle, tels l'équilibrage de charge, le basculement et les changements de types d'utilisation.

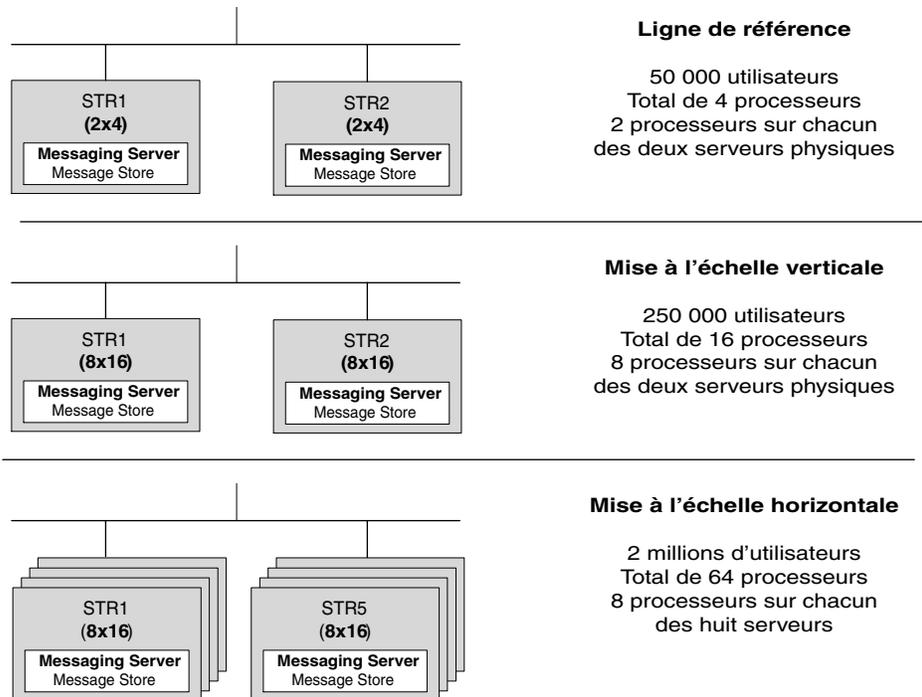


FIGURE 5-9 Exemples de mise à l'échelle horizontale et verticale

Identification des goulots d'étranglement des performances

Pour bien réussir votre déploiement, vous devez identifier les goulots d'étranglement de performances potentiels et établir une stratégie permettant de les éviter. On appelle goulot d'étranglement le moment où la vitesse d'accès aux données dépasse les exigences système spécifiées.

Les goulots d'étranglement peuvent être classés selon différentes catégories de matériel, comme le montre le tableau ci-dessous répertoriant les points d'accès aux données d'un système. Ce tableau fournit également des solutions permettant d'éviter les goulots d'étranglement pour chaque catégorie de matériel.

TABLEAU 5-7 Points d'accès aux données

Catégorie de matériel	Vitesse d'accès relative	Solutions pour améliorer les performances
Processeur	Nanosecondes	<p>Mise à l'échelle verticale : ajoutez de la puissance de traitement, augmentez le cache du processeur.</p> <p>Mise à l'échelle horizontale : ajoutez de la puissance de traitement parallèle pour l'équilibrage de charge.</p>
Mémoire système (RAM)	Microsecondes	<p>Allouez de la mémoire système à des tâches spécifiques.</p> <p>Mise à l'échelle verticale : ajoutez de la mémoire supplémentaire.</p> <p>Mise à l'échelle horizontale : créez des instances supplémentaires pour le traitement parallèle et l'équilibrage de charge.</p>
Lecture et écriture sur les disques	Millisecondes	<p>Optimisez l'accès aux disques à l'aide de baies de disques (RAID).</p> <p>Limitez l'accès aux disques à des fonctions spécifiques (lecture ou écriture seule).</p> <p>Mettez en cache les données fréquemment utilisées.</p>
Interface réseau	Varie selon la bande passante et la vitesse d'accès des nœuds du réseau.	<p>Augmentez la bande passante.</p> <p>Ajoutez du matériel d'accélération pour le transport des données sécurisées.</p> <p>Améliorez les performances des nœuds du réseau de sorte que l'accès aux données soit plus rapide.</p>

Remarque – À la section [“Identification des goulots d'étranglement des performances”](#) à la page 93, les catégories de matériel sont répertoriées en fonction de leur vitesse d'accès relative. Les points d'accès lents, tels les disques par exemple, sont plus susceptibles de provoquer des goulots d'étranglement. Toutefois, les processeurs dont la puissance ne permet pas de traiter les charges importantes représentent également des sources probables de goulots d'étranglement.

La conception du déploiement commence généralement par une estimation de la puissance de traitement requise pour chaque composant et leurs éléments dépendants. Vous déterminez ensuite la façon d'éviter les goulots d'étranglement liés à la mémoire système et à l'accès aux disques. Ensuite, vous examinez l'interface réseau pour identifier les éventuels goulots d'étranglement et élaborer des stratégies destinées à les surmonter.

Optimisation de l'accès aux disques

La vitesse d'accès aux disques contenant les données fréquemment utilisées, telles que les annuaires LDAP, est un élément essentiel de la conception du déploiement. L'accès aux disques est le mode d'accès aux données le plus lent et il constitue une source possible de goulots d'étranglement.

Pour optimiser l'accès aux disques, vous pouvez séparer les opérations d'écriture des opérations de lecture. En effet, les opérations d'écriture coûtent plus cher que les opérations de lecture et ces dernières (recherches dans les annuaires LDAP) sont beaucoup plus fréquentes que les premières (mises à jour des données des annuaires LDAP).

Vous pouvez également consacrer l'utilisation des disques à différents types d'opérations d'E/S. Par exemple, prévoyez des accès séparés pour les opérations de journalisation de Directory Server dans les journaux de transactions et d'événements, par exemple, et pour les opérations de lecture et d'écriture LDAP.

Enfin, vous pouvez envisager d'implémenter une ou plusieurs instances de Directory Server dédiées aux opérations de lecture et d'écriture et d'utiliser des instances répliquées, distribuées sur les serveurs locaux pour les accès en lecture et en recherche. Les fonctions de chaînage et de liaison sont également disponibles pour optimiser l'accès aux services d'annuaire.

Le chapitre "Defining System Characteristics" du manuel *Sun Java System Directory Server 5 2005Q1 Deployment Planning Guide* présente divers facteurs à prendre en compte lors de la planification de l'accès aux disques. Les sujets traités sont les suivants :

- **Mémoire minimale et espace disque requis** : permet d'estimer l'espace disque et la quantité de mémoire requis en fonction de la taille des annuaires.
- **Dimensionnement de la mémoire physique pour l'accès au cache** : fournit des instructions pour estimer la taille du cache en fonction de l'utilisation prévue de Directory Server et pour planifier l'utilisation totale de la mémoire.
- **Dimensionnement des sous-systèmes de disques** : fournit des informations sur la planification de l'espace disque requis en fonction du suffixe des annuaires et de facteurs liés à Directory Server influant sur l'utilisation des disques ainsi que sur la répartition des fichiers entre les disques. Présente également des solutions utilisant des baies de disques.

Conception pour une utilisation optimale des ressources

La conception du déploiement ne consiste pas simplement à évaluer les ressources nécessaires pour répondre aux exigences de qualité de service. Au cours de cette phase, vous devez également analyser toutes les solutions disponibles et choisir celle qui vous permettra de répondre aux exigences de qualité de service tout en minimisant les coûts. Vous devez analyser tous les compromis impliqués par vos décisions de conception afin de vous assurer qu'un bénéfice dans un domaine n'est pas annulé par un coût dans un autre.

Par exemple, la mise à l'échelle horizontale peut améliorer la disponibilité globale mais engendrer des coûts supplémentaires en termes de maintenance et de service. De même, la mise à l'échelle verticale peut accroître la puissance de traitement à moindres frais mais cette puissance supplémentaire risque de ne pas être utilisée de manière efficace par certains services.

Avant de mettre au point votre stratégie de conception, revoyez vos décisions pour vérifier que les avantages financiers offerts par la solution proposée compensent suffisamment l'utilisation des ressources. Au cours de cette analyse, il vous faut étudier l'impact des avantages du système dans un domaine sur les avantages du système dans d'autres domaines. Le tableau ci-dessous répertorie certains avantages du système et les éléments à prendre en compte en termes de gestion des ressources.

TABLEAU 5-8 Gestion des ressources

Qualité du système	Description
Performances	Si vous optez pour une solution dans laquelle les CPU sont rassemblés sur des serveurs individuels, les services pourront-ils utiliser efficacement la puissance de traitement ? En effet, pour certains services, le nombre de CPU pouvant être utilisés de manière efficace est limité.
Capacité latente	Votre stratégie peut-elle gérer les charges dépassant les estimations de performances ? Les charges excessives sont-elles gérées par mise à l'échelle verticale sur les serveurs, par équilibrage de charge vers d'autres serveurs ou les deux ? La capacité latente est-elle suffisante pour gérer les charges de pointe jusqu'à la prochaine étape de mise à l'échelle du déploiement ?
Sécurité	Avez-vous pris en compte les performances supplémentaires nécessaires au traitement des transactions sécurisées ?

TABLEAU 5-8 Gestion des ressources (Suite)

Qualité du système	Description
Disponibilité	Dans le cas des solutions à redondance horizontale, avez-vous estimé correctement les coûts de maintenance à long terme ? Avez-vous pris en compte les temps d'indisponibilité planifiés nécessaires aux opérations de maintenance du système ? Avez-vous comparé les coûts de serveurs haut de gamme et de serveurs d'entrée de gamme ?
Évolutivité	Avez-vous prévu des étapes de mise à l'échelle du déploiement ? Avez-vous établi une stratégie fournissant une capacité latente suffisante pour gérer les augmentations de charge prévues entre chaque étape de mise à l'échelle du déploiement ?
Entretien	Votre conception de disponibilité tient-elle compte des coûts d'administration, de contrôle et de maintenance ? Avez-vous envisagé des solutions de délégation permettant aux utilisateurs finals d'effectuer certaines tâches d'administration afin de réduire les coûts ?

Gestion des risques

La plupart des données sur lesquelles repose la conception du déploiement (exigences de qualité de service, analyse d'utilisation) ne sont pas empiriques. Il s'agit d'estimations et de projections issues des analyses d'exploitation. Bien des facteurs peuvent fausser ces projections, comme des situations imprévues liées au contexte économique, des méthodes de collecte des données inadaptées ou simplement des erreurs humaines. Avant de terminer une conception de déploiement, réexaminez les analyses sur lesquelles elle est fondée et vérifiez qu'elle prend en compte une marge d'erreur raisonnable par rapport aux estimations et aux projections.

Par exemple, si l'analyse d'utilisation sous-estime l'utilisation réelle du système, celui-ci risque d'être incapable de faire face au trafic auquel il est soumis. Une conception dont les performances ne sont pas satisfaisantes sera probablement considérée comme un échec.

A contrario, si la puissance de votre système est supérieure aux exigences, des ressources qui pourraient être utilisées à d'autres fins sont monopolisées inutilement. La bonne méthode consiste à prévoir une marge de sécurité suffisante tout en utilisant les ressources à bon escient.

Une utilisation inadaptée des ressources entraîne un échec de la conception car les ressources sous-utilisées auraient pu être employées pour d'autres activités. En outre, des solutions jugées inappropriées risquent d'être considérées par les parties prenantes comme ne respectant pas les termes du contrat.

Exemple d'architecture de déploiement

Le schéma ci-dessous représente une architecture de déploiement complète correspondant à l'exemple décrit plus haut dans ce document. Il donne un aperçu de la manière dont une architecture de déploiement peut se présenter.



Attention – L'architecture de déploiement illustrée par le schéma ci-dessous est fournie à titre d'exemple uniquement. Il ne s'agit pas d'un déploiement ayant été conçu, mis en place et testé en conditions réelles et ne doit donc pas être considéré comme une recommandation.

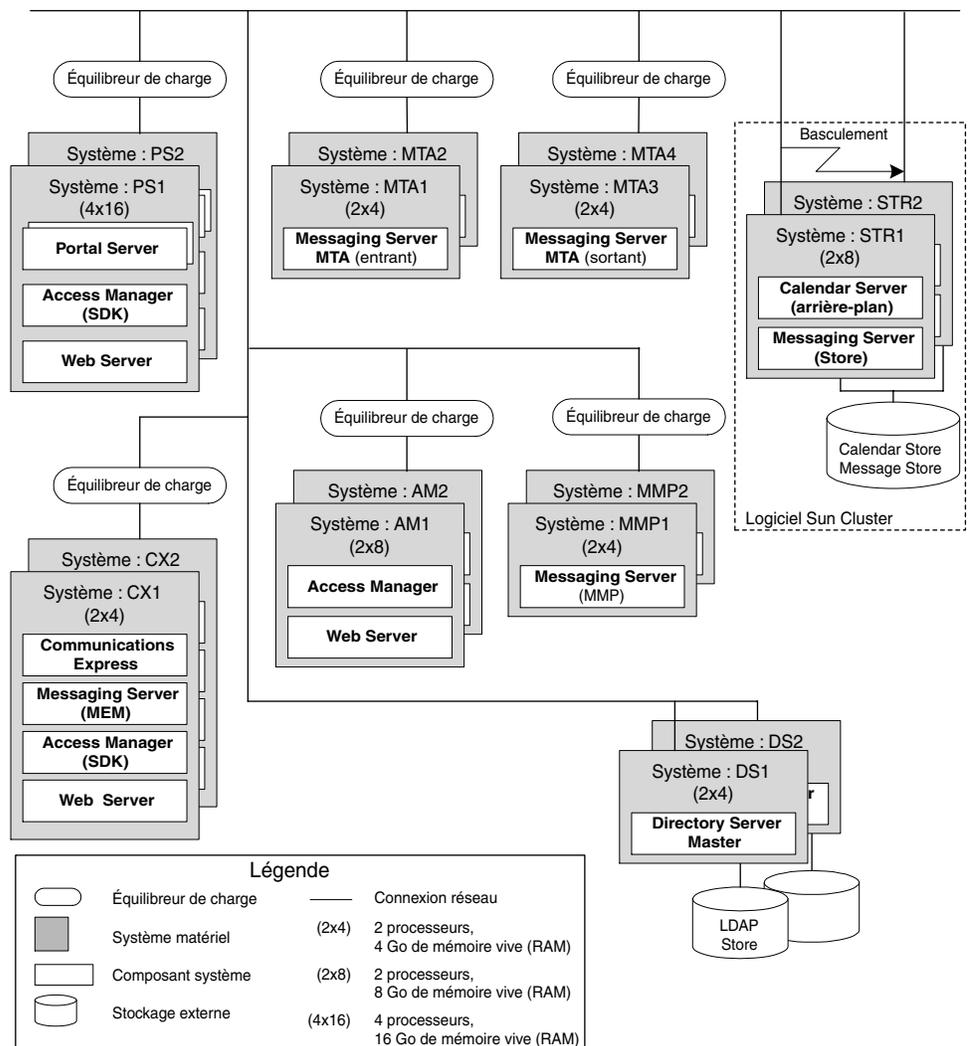


FIGURE 5-10 Exemple d'architecture de déploiement

Implémentation d'une conception de déploiement

Dans le cycle de vie de la solution, lors de la phase d'implémentation, vous partez des spécifications et des plans créés au cours de la conception du déploiement pour mettre en place et tester l'architecture de déploiement et, en dernier lieu, pour la déployer en environnement de production. Bien que l'objet de ce manuel ne soit pas de fournir une description détaillée de la phase d'implémentation, vous en trouverez une présentation générale dans ce chapitre.

Ce chapitre se compose des sections suivantes :

- "À propos de l'implémentation des conceptions de déploiement" à la page 101
- "Installation et configuration du logiciel" à la page 102
- "Développement de pilotes et de prototypes" à la page 103
- "Test des pilotes et des prototypes" à la page 103
- "Passage à un déploiement de production" à la page 104

À propos de l'implémentation des conceptions de déploiement

Une fois l'architecture de déploiement approuvée et les spécifications et les plans d'implémentation créés, il est temps de passer à la phase d'implémentation. L'implémentation est le résultat d'un ensemble complexe de processus et de procédures dont la réussite passe par une préparation minutieuse. Les tâches d'implémentation sont les suivantes :

- élaboration de l'infrastructure réseau et matérielle ;
- installation et configuration des logiciels conformément au plan d'installation ;
- migration des données des applications existantes vers la solution en cours ;
- implémentation d'un plan de gestion des utilisateurs

- conception et déploiement de pilotes ou de prototypes dans un environnement de test conformément à un plan de test ;
- conception et exécution de tests fonctionnels et de tests de charge en fonction d'un plan de test ;
- déploiement de la solution d'un environnement de test dans un environnement de production conformément à un plan de déploiement ;
- formation des administrateurs et des utilisateurs du déploiement selon un plan de formation.

L'objet de ce manuel n'est pas de fournir une description détaillée de l'implémentation. Toutefois, les sections qui suivent proposent une présentation de certaines tâches relatives à l'implémentation.

Installation et configuration du logiciel

L'installation et la configuration de Sun Java™ Enterprise System pour une application d'entreprise distribuée impliquent la planification et la coordination de nombreuses tâches et procédures. Au cours de la phase de conception du déploiement, vous devez créer un plan d'installation fondé sur l'architecture de déploiement de haut niveau qui fournit les informations de configuration requises pour installer Java Enterprise System.

Ce plan d'installation décrit les opérations suivantes :

- identification de la séquence et du type d'installation ;
- vérification des logiciels installés sur les hôtes et de la capacité de ces derniers à accueillir une installation ;
- collecte des informations de configuration relatives à chaque composant de Java Enterprise System à installer.

Le manuel *Guide de planification de l'installation de Java ES System 2005Q4* comprend des indications détaillées sur la manière de rassembler les informations nécessaires à un plan d'installation. Le manuel *Référence de l'installation de Java ES System 2005Q4* fournit des informations de configuration détaillées et des feuilles de travail complétant ces informations. Le manuel *Guide d'installation de Sun Java Enterprise System 2005Q4 pour UNIX* décrit les scénarios d'installation courants impliquant plusieurs composants de Java Enterprise System. Pour plus d'informations, reportez-vous au Chapitre 1, "Préparation de l'installation" du *Guide d'installation de Sun Java Enterprise System 2005Q4 pour UNIX*.

Développement de pilotes et de prototypes

Il existe deux catégories de déploiement Java Enterprise System : ceux qui font appel principalement aux services fournis par Java Enterprise System et ceux qui utilisent de nombreux services personnalisés intégrés aux services de Java Enterprise System. Le premier déploiement mentionné est du type « 80-20 » (80 % des services sont fournis par Java Enterprise System) et le second, du type « 20-80 ».

Lors de la phase d'implémentation des déploiements du type 80-20, vous développez généralement un pilote à des fins de test. Les déploiements de ce type faisant appel à des services Java Enterprise System complets offrant des fonctionnalités prêtes à l'emploi, le passage des pilotes de la phase de développement, de test et de modification à la phase de déploiement de la production s'effectue assez rapidement. Les pilotes sont destinés à vérifier le fonctionnement d'une solution et à fournir des informations sur les performances du système.

Les déploiements de type 20-80, quant à eux, font appel à des services nouveaux et personnalisés qui n'ont pas fait la preuve de leur interopérabilité. C'est la raison pour laquelle vous devez créer un prototype destiné à démontrer le bien-fondé du déploiement. La mise en production de ce prototype exige un cycle de développement, de test et de modification particulièrement rigoureux. Un prototype vous permet de déterminer si la solution proposée résout correctement le problème voulu dans un environnement de test. Lorsque vous jugez que le prototype fonctionne selon vos attentes, vous pouvez le soumettre à des tests plus approfondis avant de le faire passer à l'état de pilote.

Remarque – Le pourcentage de services personnalisés requis peut varier considérablement d'un déploiement d'entreprise à un autre. La façon dont vous utilisez vos pilotes et vos prototypes dépend de la complexité et de la nature de votre déploiement.

Test des pilotes et des prototypes

Tester les pilotes et les prototypes permet de déterminer le plus précisément possible si, dans des conditions de test, un déploiement satisfait les exigences système et atteint les objectifs d'exploitation.

Dans l'idéal, les tests fonctionnels doivent s'inspirer des scénarios fondés sur les cas d'utilisation identifiés et un ensemble de mesures permettant d'évaluer la conformité des performances doit être créé. Le test fonctionnel peut également consister à effectuer un déploiement limité auprès d'un groupe réduit d'utilisateurs afin de déterminer si les exigences d'exploitation sont satisfaites.

Les tests de charge permettent d'évaluer les performances du système en situation de charge de pointe. Ils font généralement appel à des environnements simulés et à des générateurs de charge pour mesurer la capacité de traitement des données et les performances. Les exigences système relatives au déploiement sont utilisées comme base pour la conception et la réussite des tests de charge.

Remarque – Les tests fonctionnels et les tests de charge sont essentiels dans le cas de déploiements de grande envergure pour lesquels les exigences système sont susceptibles de ne pas être définies très précisément, qui n'ont pas fait l'objet d'une implémentation précédente sur laquelle fonder les estimations et qui exigent un important travail de développement.

Les tests peuvent mettre en évidence des problèmes liés aux spécifications de conception du déploiement et impliquer plusieurs phases de conception, de création et de test avant que le déploiement puisse passer en phase de production. Les tests des prototypes peuvent révéler des problèmes de conception, qui vous obligeront à revenir plusieurs étapes en arrière dans le cycle de vie de la solution.

Avant de passer à un déploiement pilote, assurez-vous que vous avez soumis votre conception à des tests approfondis. En effet, ce type de déploiement implique que les tests de la conception ont donné satisfaction. Les problèmes détectés lors du test d'un déploiement pilote doivent être résolus dans les paramètres de la conception du déploiement.

Étant donné que les tests ne permettent pas une simulation parfaite d'un environnement de production et parce que la nature d'une solution déployée peut évoluer et changer, vous devez soumettre les systèmes déployés à des contrôles afin d'identifier les éléments devant faire l'objet d'opérations de maintenance, d'adaptation ou d'entretien.

Passage à un déploiement de production

Une fois que le pilote ou le déploiement a fait la preuve du bien-fondé de sa conception en satisfaisant les critères de test, vous pouvez le faire passer en environnement de production. Cette opération s'effectue généralement de façon progressive. L'approche progressive est particulièrement importante pour les déploiements de grande ampleur qui touchent de nombreux utilisateurs.

Vous pouvez d'abord déployer la solution auprès d'un nombre réduit d'utilisateurs, puis augmenter ce nombre jusqu'à ce que le déploiement soit complet. Vous pouvez également commencer par déployer un petit nombre de services avant de continuer le déploiement et d'intégrer les services restants. Cette méthode permet d'isoler, d'identifier et de résoudre les problèmes auxquels un service peut être confronté dans un environnement de production.

Index

Nombres et symboles

- 20-80
 - Déploiement, 19
 - Implémentation, phase, 103
- 80-20
 - Déploiement, 19, 103

A

- Access Manager, 53, 77
- Accords de niveau de service, Impact sur la conception du déploiement, 72
- Analyse d'exploitation, phase, 22
 - À propos, 27
- Analyse d'utilisation, 36-38
 - Impact sur la conception du déploiement, 72
- Application Server, 53
- Approbation du projet, 70
- Architecture à 3 dimensions, 50
- Architecture à plusieurs niveaux, conception, 56-58
- Architecture à trois dimensions, 50
- Architecture de déploiement, 70-71
 - Exemple, 98-99
- Architecture logique, 49-50
 - Conception, 51
 - Exemple, 58-67
 - Exemple de communications basées sur les identités, 63
 - Impact sur la conception du déploiement, 71-72

B

- Basculement, 83
 - Exemple, 88-89
 - Logiciel Sun Cluster, 86

C

- Calendar Server, 53, 77
- Capacité latente, 46
 - Évolutivité, 92
- Cas d'utilisation, 38
 - Estimation de la puissance de traitement, 77-78
 - Exemple de communications basées sur les identités, 64-67
 - Exemple de Messaging Server, 60-63
 - Impact sur la conception du déploiement, 72
- Communications basées sur les identités, exemple, 63-67
 - Cas d'utilisation, 64-67
- Communications Express, 53
- Conception du déploiement
 - À propos, 70-72
 - Approbation du projet, 70
 - Facteurs, 71-72
 - Méthodologie, 73-74
 - Résultats, 70-71
- Conception du déploiement, phase, 23-24
- Conception logique, À propos, 49-50
- Conception logique, phase, 23
- Contraintes d'exploitation, 33-34
 - Coût de possession, 34

- Contraintes d'exploitation (Suite)
 - Établissement de la planification, 33
 - Limitations budgétaires, 33-34
 - Problème de migration, 33
- Contrat de niveau de service, 32
- Contrats de niveau de service, Exigences, 47
- Coût de possession, 34
- Coût total de possession, Impact sur la conception du déploiement, 72
- Culture d'entreprise, 30-32
- Cycle de vie de la solution, 20-22
 - Phase d'analyse d'exploitation, 22, 27
 - Phase d'implémentation, 24, 101-102
 - Phase de conception du déploiement, 23-24, 70-72
 - Phase de conception logique, 23, 49-50
 - Phase de fonctionnement, 24-25
 - Phase de spécification technique, 22-23, 35-36

D

- Dépendances entre composants, 52-55
 - Prise en charge de conteneur Web, 55
- Directory Proxy Server, 53, 56
- Directory Server, 53, 59, 77
 - Réplication monomaître, 89
 - Réplication multimaître, 89, 90-91
- Disponibilité
 - Basculement, 83
 - Classement par ordre de priorité, 42
 - Équilibrage de charge, 83
 - Exemple, 87-91
 - Exigence de qualité de service, 40-42
 - Optimisation des ressources, 97
 - Réplication des services, 83
 - Système à basculement N+1, 84-86
 - Système à redondance horizontale, 84-86
- DMZ, Zone d'accès externe, 68
- Documentation
 - Guide d'installation, 53, 102
 - Présentation technique, 18, 50, 51

E

- Entretien
 - Exigence de qualité de service, 46-47
 - Optimisation des ressources, 97
- Équilibrage de charge, 83
 - Exemple, 85
- Estimation de la puissance de traitement, 73, 74-79
 - Cas d'utilisation, 77-78
 - Exemple, 75-79
 - Transactions sécurisées, 80-82
- Établissement de la planification, 33
- Évolutivité
 - Estimation du développement, 43-44
 - Exemple, 92-93
 - Exigence de qualité de service, 43-44
 - Optimisation des ressources, 97
 - Stratégies, 91-93
- Exemple
 - Architecture de déploiement, 98-99
 - Architecture logique, 58-67
 - Architecture logique de Messaging Server, 58-63
 - Basculement, 88-89
 - Communications basées sur les identités, 63-67
 - Conception de disponibilité, 87-91
 - Équilibrage de charge, 85, 87-88
 - Estimation de la puissance de traitement, 75-79
 - Estimation du nombre de processeurs pour les transactions sécurisées, 80-82
 - Évolutivité, 92-93
 - Réplication des services, 89
 - Réplication monomaître, 89-90
 - Réplication multimaître, 90-91
 - Serveur d'annuaire, 89
 - Zone d'accès, 67
- Exemple de communications basées sur les identités, Estimation de la puissance de traitement, 75
- Exigences d'entreprise, Type d'utilisation, 30
- Exigences de l'entreprise
 - Contrat de niveau de service, 32
 - Culture d'entreprise, 30-32
 - Définition, 28-32
 - Exigences opérationnelles, 29-30
 - Exigences réglementaires, 31

- Exigences de l'entreprise (Suite)
 - Identification des besoins des utilisateurs, 29-30
 - Objectifs d'exploitation, 28-29
 - Objectifs de sécurité, 31
- Exigences de niveau de service, 47
- Exigences de qualité de service, 39-47
 - Impact sur la conception du déploiement, 72
 - Rôle dans la conception du déploiement, 70
- Exigences opérationnelles, 29-30
- Exigences réglementaires, 31
- Exigences techniques
 - Capacité latente, 46
 - Disponibilité, 40-42
 - Entretien, 46-47
 - Évolutivité, 43-44
 - Exigences de niveau de service, 47
 - Performances, 40
 - Sécurité, 44-45

F

Fonctionnement, phase, 24-25

G

- Gestion des risques, 97-98
 - Conception du déploiement, 74
- Glossaire, lien, 13

I

- Identification des goulots d'étranglement,
 - Conception du déploiement, 73
- Implémentation, phase, 24, 103
 - À propos, 101-102
 - Développement de pilotes et de prototypes, 103
- Instant Messaging, 53

J

- Java Enterprise System
 - 20-80
 - Déploiement, 19
 - 80-20
 - Déploiement, 19
 - À propos, 17
 - Architecture à trois dimensions, 50
 - Composants, 51-58
 - Composants d'accès, 56
 - Dépendances entre composants, 52-55
 - Installation, 102
 - Passage à un déploiement de production, 104-105
 - Problèmes de migration, 19-20
 - Service, 19
 - Services personnalisés, 19
 - Services système, 17-18
- Java Enterprise System, installation, 102

L

Limitations budgétaires, 33-34

M

- Manuel d'exploitation, 71
- Message Queue, 53
- Messaging Server, 53
 - Cas d'utilisation, 60-63
 - Exemple d'architecture logique, 58-63
 - Exemple d'équilibrage de charge, 87-88
 - Message Multiplexor (MMP), 55, 56, 59, 77
 - Message Store (STR), 55, 59, 77
 - Message Transfer Agent (MTA), 55, 59
 - Messenger Express Multiplexor (MEM), 55
 - Services logiques distincts, 55-56
- Migration, problème, Contraintes d'exploitation, 33
- Migration, problèmes, 19-20

N

Niveau client, Modèle d'architecture à plusieurs niveaux, 57

- Niveau données, Modèle d'architecture à plusieurs niveaux, 57
- Niveau logique, Modèle d'architecture à plusieurs niveaux, 57
- Niveau présentation, Modèle d'architecture à plusieurs niveaux, 57
- Niveau services d'entreprise, Modèle d'architecture à plusieurs niveaux, 57

O

- Objectifs d'exploitation
 - Définition, 28-29
 - Impact sur la conception du déploiement, 72
- Optimisation
 - Accès aux disques, 95
 - Utilisation des ressources, 96-97
- Optimisation des ressources, Conception du déploiement, 73

P

- Performances
 - Exigence de qualité de service, 40
 - Identification des goulots d'étranglement, 93-95
 - Optimisation des ressources, 96
- Pilote, 103
 - Test, 103-104
- Plan d'installation, 71
- Plan de déploiement, 71
- Plan de fonctionnement (manuel d'exploitation), 71
- Plan de formation, 71
- Plan de gestion des utilisateurs, 71
- Plan de migration, 71
- Plan de reprise sur sinistre, 71
- Plan de test, 71
- Planification du déploiement
 - À propos, 20-25
 - Approche incrémentielle, 32
 - Cycle de vie de la solution, 20-22
- Plans d'implémentation, 70-71
- Portal Server, 53, 56
 - Mobile Access, 56
 - Secure Remote Access, 53, 56

- Prototype, 103
 - Test, 103-104
- Puissance de traitement, Estimation, 74-79

Q

- QoS (exigences de qualité de service), 39-47

R

- Réplication de services, 73
- Réplication des services
 - Exemple du serveur d'annuaire, 89
 - Stratégie de disponibilité, 83
- Réplication monomaître, 89
 - Exemple, 89-90
- Réplication multimaître, 89
 - Exemple, 90-91
- Risques, gestion, 97-98

S

- Scénario de déploiement, 68, 70
- scénario de déploiement, 49
- Sécurité
 - Estimation de la puissance de traitement requise, 73
 - Exigence de qualité de service, 44-45
 - Optimisation des ressources, 96
- SLA, 32
- Spécification technique, phase, 22-23
 - À propos, 35-36
 - Analyse d'utilisation, 36-38
 - Cas d'utilisation, 38
 - Exigences de qualité de service, 39-47
- Spécifications d'implémentation, 70-71
- Stratégies de disponibilité, Identification, 83-91
- Sun Cluster, logiciel, 86
 - Exemple de basculement, 88-89
- Système à basculement N+1, 84-86
- Système à redondance horizontale, 84-86
- Systèmes à tolérance de pannes, 41

T

Test

- Pilotes et prototypes, 103-104
- Test de charge, 104
- Test fonctionnel, 104
- Test de charge, 104
- Test fonctionnel, 104

U

- Utilisation, type, 30

W

- Web Server, 53, 77

Z

- Zone d'accès, 67-68
- Zone d'accès externe (DMZ), 68
- Zone d'accès interne (intranet), 68
- Zone d'accès sécurisé, 68

