



Sun Java™ System

Guide de planification du déploiement Sun Java Enterprise System 2005T1

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

Référence : 819-1918

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

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

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

Cette distribution peut comprendre des composants développés par des tierces parties.

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

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

Toutes les marques SPARC sont utilisées 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.

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

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

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

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

Sommaire

Liste des figures	7
Liste des tableaux	9
Préface	11
À qui ce manuel s'adresse-t-il ?	12
Organisation du manuel	12
Documentation connexe	12
Manuels composant cette documentation	12
Accès aux ressources Sun en ligne	14
Comment obtenir une assistance technique	15
Références aux sites Web tiers associés	15
Sun attend vos commentaires	15
Chapitre 1 Introduction à la planification du déploiement	17
À propos de Java Enterprise System	17
Services système	18
Services intégrés et services personnalisés	20
Migration vers Java Enterprise System	21
À propos de la planification du déploiement	21
Cycle de vie de la solution	22
Phase d'analyse d'exploitation	23
Phase d'exigences techniques	23
Phase de conception logique	24
Phase de conception du déploiement	25
Phase d'implémentation	25
Phase de fonctionnement	26

Chapitre 2 Analyse d'exploitation	27
A propos de l'analyse d'exploitation	27
Définition des exigences de l'entreprise	28
Définition des objectifs d'exploitation	28
Portée	28
Priorités	28
Qualités essentielles	28
Facteurs de croissance	29
Marge de sécurité	29
Identification des besoins des utilisateurs	29
Développement des exigences opérationnelles	30
Prise en charge des modèles d'utilisation existants	30
Connaissance de la culture de l'entreprise	31
Parties prenantes	31
Normes et stratégies	31
Exigences réglementaires	31
Sécurité	32
Répartition géographique	32
Approche incrémentielle	32
Contrats de niveau de service (SLA)	33
Définition de contraintes d'exploitation	33
Problèmes de migration	34
Établissement du planning	34
Limitations budgétaires	34
Coût de propriété	35
Chapitre 3 Exigences techniques	37
À propos des exigences techniques	37
Analyse d'utilisation	38
Cas d'utilisation	40
Exigences de qualité de service	41
Performances	42
Disponibilité	43
Systèmes à tolérance de pannes	44
Classement par ordre de priorité de la disponibilité du service	44
Panne de services	45
Évolutivité	45
Estimation du développement	46
Sécurité	47
Éléments d'un plan de sécurité	48
Capacité latente	49
Exigences d'entretien	49
Exigences de niveau de service	51

Chapitre 4 Conception logique	53
À propos des architectures logiques	53
Conception d'une architecture logique	55
Composants de Java Enterprise System	56
Dépendances entre composants	57
Support de conteneur Web	60
Services logiques distincts fournis par Messaging Server	60
Composants d'accès	61
Conception d'architecture à plusieurs niveaux	61
Exemples d'architectures logiques	63
Exemple Messaging Server	63
Cas d'utilisation de Messaging Server	65
Exemple de communications basées sur les identités	68
Cas d'utilisation pour l'exemple de communications basées sur les identités	69
Zones d'accès	72
Scénario de déploiement	73
Chapitre 5 Conception du déploiement	75
À propos de la conception du déploiement	76
Approbation du projet	76
Résultats de la conception du déploiement	76
Facteurs affectant la conception du déploiement	78
Méthodologie de conception du déploiement	79
Estimation de la puissance de traitement	80
Exemple d'estimation de la puissance de traitement	82
Estimation du nombre de processeurs requis pour les points d'entrée utilisateur	82
Inclusion du nombre de processeurs estimé pour les services dépendants	84
Étude des cas d'utilisation pour les charges de pointe	85
Modification des estimations pour les autres conditions de charge	85
Mise à jour du nombre de processeurs estimé	86
Estimation de la puissance de traitement requise pour les transactions sécurisées	87
Estimation du nombre de processeurs pour les transactions sécurisées	88
Matériel dédié au traitement des transactions SSL	90
Identification des stratégies de disponibilité	91
Stratégies de disponibilité	91
Système à serveur unique	92
Systèmes à redondance horizontale	93
Logiciel Sun Cluster	95

Exemples de conceptions de disponibilité	95
Exemple d'équilibrage de charge pour Messaging Server	95
Exemple de basculement à l'aide du logiciel Sun Cluster	97
Exemple de réplication des services d'annuaire	97
Réplication monomaître	98
Réplication multimaître	99
Identification des stratégies d'évolutivité	100
Capacité latente	100
Exemple d'évolutivité	101
Identification des goulots d'étranglement des performances	102
Optimisation de l'accès aux disques	104
Conception pour une utilisation optimale des ressources	105
Gestion des risques	107
Exemple d'architecture de déploiement	107
Chapitre 6 Implémentation d'une conception de déploiement	109
À propos de l'implémentation des conceptions de déploiement	110
Installation et configuration du logiciel	111
Développement de pilotes et de prototypes	111
Test des pilotes et des prototypes	112
Passage à un déploiement de production	113
Index	115

Liste des figures

Figure 1-1	Cycle de vie de la solution	22
Figure 4-1	Trois dimensions de l'architecture de la solution Java Enterprise System	54
Figure 4-2	Composants de Java Enterprise System	56
Figure 4-3	Dépendances entre composants de Java Enterprise System	59
Figure 4-4	Modèle d'architecture à plusieurs niveaux	62
Figure 4-5	Architecture logique de déploiement de Messaging Server	64
Figure 4-6	Architecture logique de Messaging Server illustrant le premier cas d'utilisation ...	66
Figure 4-7	Architecture logique de Messaging Server illustrant le deuxième cas d'utilisation ..	67
Figure 4-8	Architecture logique de Messaging Server illustrant le troisième cas d'utilisation ..	68
Figure 4-9	Architecture logique d'un scénario de communications basées sur les identités	69
Figure 4-10	Architecture logique de scénario de communications illustrant le premier cas d'utilisation	70
Figure 4-11	Architecture logique de scénario de communications illustrant le deuxième cas d'utilisation	71
Figure 4-12	Composants logiques placés dans des zones d'accès	72
Figure 5-1	Architecture logique d'un scénario de communications basées sur les identités	83
Figure 5-2	Système à serveur unique	92
Figure 5-3	Système à basculement N+1 avec deux serveurs	93
Figure 5-4	Équilibrage de charge et basculement entre deux serveurs	93
Figure 5-5	Répartition de la charge entre n serveurs	94
Figure 5-6	Stratégie de disponibilité pour l'exemple Messaging Server	96
Figure 5-7	Conception avec basculement utilisant le logiciel Sun Cluster	97
Figure 5-8	Exemple de réplication monomaître	98
Figure 5-9	Exemple de réplication multimaître	99
Figure 5-10	Exemples de mise à l'échelle horizontale et verticale	102
Figure 5-11	Exemple d'architecture de déploiement	108

Liste des tableaux

Tableau 1	Documentation Java Enterprise System	13
Tableau 1-1	Catégories de services Java Enterprise System	19
Tableau 3-1	Paramètres à prendre en considération lors de l'analyse d'utilisation	39
Tableau 3-2	Qualités systèmes ayant un impact sur les exigences de qualité de service	41
Tableau 3-3	Interruption d'activité non planifiée d'un système fonctionnant toute l'année (8 760 heures)	44
Tableau 3-4	Disponibilité des services par ordre de priorité	45
Tableau 3-5	Facteurs d'évolutivité	47
Tableau 3-6	Paramètres concernant les exigences d'entretien	49
Tableau 4-1	Java Enterprise System Dépendances entre composants de	57
Tableau 4-2	Messaging Server Configurations	60
Tableau 4-3	Composants de Java Enterprise System fournissant un accès distant	61
Tableau 4-4	Niveaux logiques dans une architecture à plusieurs niveaux	62
Tableau 4-5	Composants d'une architecture logique de Messaging Server	64
Tableau 4-6	Zones d'accès sécurisé et les composants qu'elles contiennent	73
Tableau 5-1	Estimation du nombre de processeurs pour les composants contenant des points d'entrée utilisateur	83
Tableau 5-2	Estimation du nombre de processeurs pour les composants de support	84
Tableau 5-3	Modification de l'estimation du nombre de processeurs pour les charges de pointe	85
Tableau 5-4	Modification de l'estimation du nombre de processeurs pour les composants de support	86
Tableau 5-5	Modification du nombre de processeurs estimé pour les transactions sécurisées	89
Tableau 5-6	Modification du nombre de processeurs estimé pour les transactions Portal Server sécurisées	90
Tableau 5-7	Modification de l'estimation du nombre de processeurs pour les composants de support	95
Tableau 5-8	Points d'accès aux données	103
Tableau 5-9	Considérations relatives à la gestion des ressources	106

Préface

Le Guide de planification du déploiement de Sun Java Enterprise System 2005T1 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 conception et de la planification de déploiement, traite du cycle de vie d'une solution, qui englobe les 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.

Cette préface se compose des sections suivantes :

- « À qui ce manuel s'adresse-t-il ? », page 12
- « Documentation connexe », page 12
- « Accès aux ressources Sun en ligne », page 14
- « Comment obtenir une assistance technique », page 15
- « Références aux sites Web tiers associés », page 15
- « Sun attend vos commentaires », page 15

Avant d'effectuer les tâches décrites dans ce guide, lisez le document *Sun Java Enterprise System 2005T1 - Notes de version* (<http://docs.sun.com/doc/819-0814>).

À qui ce manuel s'adresse-t-il ?

Il est principalement destiné aux architectes 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.

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 2005T1*.

Organisation du manuel

Ce manuel est fondé sur un cycle de vie de solution décrivant les différentes phases de la planification du déploiement. Le [chapitre 1, « Introduction à la planification du déploiement »](#), décrit le cycle de vie de la solution.

Documentation connexe

Le site Web <http://docs.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 manuel ou un sujet spécifique.

Manuels composant cette documentation

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

<http://docs.sun.com>

La documentation relative à Java ES inclut des informations sur le système global de même que des informations sur chacun de ses composants. Pour y accéder, entrez l'adresse URL ci-après :

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

Le tableau suivant répertorie les manuels relatifs au système faisant partie de la documentation de Java ES. La colonne de gauche indique le nom et le numéro de chaque document, et la colonne de droite décrit le contenu général du document.

Tableau 1 Documentation Java Enterprise System

Document	Contenu
<i>Sun Java Enterprise System 2005T1 - Notes de version</i> http://docs.sun.com/doc/819-0814	Ce manuel contient les informations les plus récentes sur Java Enterprise System, y compris les problèmes connus. De plus, chaque composant dispose de notes de version qui lui sont propres.
<i>Présentation de la documentation de Sun Java Enterprise System 2005T1</i> http://docs.sun.com/doc/819-1911	Ce guide décrit la documentation relative à Java Enterprise System. Il comporte des liens renvoyant à la documentation associée aux composants.
<i>Présentation technique de Sun Java Enterprise System 2005T1</i> http://docs.sun.com/doc/819-1925	Ce guide décrit les principes techniques et conceptuels de Java Enterprise System. Il décrit les composants, l'architecture, les processus et les fonctions.
<i>Guide de planification du déploiement de Sun Java Enterprise System 2005T1</i> http://docs.sun.com/doc/819-1918	Ce manuel indique brièvement les modalités de planification et de conception des solutions de déploiement d'entreprise basées sur Java Enterprise System. Il présente quelques principes et concepts fondamentaux de la planification et de la conception de déploiement, traite du cycle de vie d'une solution et fournit des exemples et des stratégies de haut niveau à utiliser lors de la planification de solutions basées sur Java Enterprise System.
<i>Guide de gestion des utilisateurs de Sun Java Enterprise System 2005T1</i> http://docs.sun.com/doc/817-5761	Ce document vous aide à planifier, déployer et gérer les informations relatives aux utilisateurs de votre solution Java Enterprise System. Il complète le manuel <i>Guide de planification du déploiement de Sun Java Enterprise System 2005T1</i> en décrivant les problèmes de gestion utilisateur dans chaque phase du cycle de vie d'une solution.
<i>Sun Java Enterprise System 2003T4 - Série d'exemples de déploiement : Scénario d'évaluation</i> http://docs.sun.com/doc/819-0059	Ce manuel décrit les procédures d'installation de Java Enterprise System sur un ordinateur, de mise en place des principaux services réseau partagés et de configuration des comptes utilisateur pouvant accéder à ces services.
<i>Guide d'installation de Sun Java Enterprise System 2005T1</i> http://docs.sun.com/doc/819-0807	Ce manuel vous guide tout au long de la procédure d'installation de Java Enterprise System pour le système d'exploitation Solaris™ ou Linux. Il vous explique comment sélectionner les composants à installer, les configurer et comment vous assurer de leur fonctionnement correct.

Tableau 1 Documentation Java Enterprise System (*Suite*)

Document	Contenu
<i>Guide de migration et de mise à niveau de Sun Java Enterprise System 2005T1</i> http://docs.sun.com/doc/819-0062	Ce manuel fournit des informations et des instructions sur la mise à niveau de Java Enterprise System pour l'environnement d'exploitation Solaris™ ou Linux.
<i>Glossaire de Sun Java Enterprise System</i> http://docs.sun.com/doc/816-6873	Ce document définit les termes utilisés dans la documentation de Java Enterprise System.

Accès aux ressources Sun en ligne

Pour télécharger des produits, obtenir des services professionnels, des patches, une assistance technique et des informations destinées aux développeurs, visitez les sites suivants :

- Centre de téléchargement
<http://www.sun.com/software/download/>
- Services professionnels
<http://www.sun.com/service/sunps/sunone/index.html>
- Services d'entreprise Sun, patches Solaris et support technique
<http://sunsolve.sun.com/>
- Informations pour les développeurs
<http://developers.sun.com/prodtech/index.html>

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

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

Comment obtenir une assistance technique

Pour toute question technique relative à ce produit non couverte par la documentation, rendez-vous sur <http://www.sun.com/service/contacting>.

Références aux sites Web tiers associés

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

Sun attend vos commentaires

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

Pour nous faire part de vos commentaires, rendez-vous sur le site <http://docs.sun.com>, puis cliquez sur Envoyer des commentaires. Dans le formulaire en ligne, indiquez le titre et le numéro du document. Ce numéro est constitué de sept ou neuf chiffres et figure sur la page de titre du manuel ou en haut du document.

Sun attend vos commentaires

Introduction à la planification du déploiement

Ce chapitre présente une brève description de Sun Java™ Enterprise System (Java ES), aborde les concepts de planification du déploiement et présente le cycle de vie de la solution, qui énumère les étapes de planification et de conception des systèmes logiciels d'entreprise. Ce chapitre se compose des sections suivantes :

- « À propos de Java Enterprise System », page 17
- « À propos de la planification du déploiement », page 21

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 en outre un système intégré de gestion des identités utilisateur et de la sécurité.

Services système

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

- **Services de portail** : ils permettent aux itinérants, aux télétravailleurs, aux professionnels du savoir, 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 extérieur au réseau d'entreprise. Ces services garantissent un accès aux groupes d'utilisateurs, 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 d'agendas sont des exemples de capacités particulières applicables à l'environnement de travail de l'utilisateur.
- **Services de sécurité et d'identité réseau** : ces services améliorent la sécurité et la protection des ressources d'informations clés de l'entreprise en garantissant l'application de stratégies de contrôle d'accès appropriées dans toutes les communautés; toutes les applications et tous les services à un niveau général. 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 pratiquement 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 des services à chaque catégorie.

Tableau 1-1 Catégories de services Java Enterprise System

Catégorie de service	Composants de 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, consultez le document *Présentation technique de Sun Java Enterprise System 2005T1*, à l'adresse <http://docs.sun.com/doc/819-1925>.

Services intégrés et services personnalisés

Les solutions de déploiement fondées sur Java Enterprise System se classent dans deux catégories générales :

- **Déploiements 80:20** : ces solutions font principalement appel à des services fournis par Java Enterprise System. Java Enterprise System fournit au moins 80 % des services.
- **Déploiements 20:80** : ces solutions font appel à de nombreux services personnalisés et applications tierces.

80:20 et 20:80 sont des catégories générales. Le pourcentage exact du type de services offerts n'est pas important. Toutefois, il indique la proportion de personnalisation d'une solution.

Java Enterprise System est particulièrement adapté aux déploiements 80:20 en raison du large éventail de services assurés par Java ES. 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 répertoriées dans le [Tableau 1-1, page 19](#), peuvent être utilisées dans le cadre de déploiements 80:20. Par exemple, les services de communication et de collaboration fournissent les services de messagerie, d'agenda 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 une adaptation des services de plate-forme J2EE peuvent tirer parti 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 un 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. Dans le cas d'une entreprise qui envisage une solution de déploiement pour la première fois, la planification, la conception et l'implémentation sont déterminées en grande partie par les besoins propres à l'entreprise. Toutefois, les premières solutions de déploiement ne sont pas les plus fréquentes. Celles qui utilisent Java Enterprise System pour consolider les solutions d'entreprise existantes ou pour effectuer une mise à niveau des composants de Java Enterprise System sont plus nombreuses.

Lors du remplacement ou de la mise à niveau de solutions existantes, vous devez passer par des étapes supplémentaires de planification, de conception et d'implémentation pour vous assurer que les données existantes sont préservées et que les logiciels sont correctement mis à niveau 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 en cours de Java Enterprise System et sur les stratégies de migration à partir d'autres applications, consultez le *Guide de migration et de mise à niveau de Sun Java Enterprise System*.

À propos de la planification du déploiement

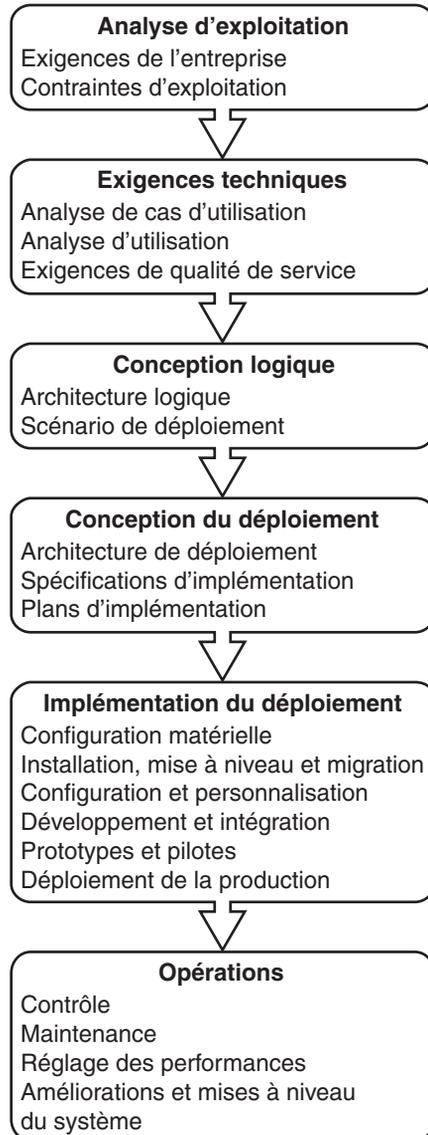
La planification du déploiement est une étape essentielle de l'implémentation d'une solution Java Enterprise System. Chaque entreprise définit ses propres objectifs, exigences et priorités. Une planification réussie commence par l'analyse des objectifs de l'entreprise et des exigences requises pour atteindre ces objectifs. Les exigences de l'entreprise doivent ensuite être converties en 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 qui n'a pas été convenablement planifié peut connaître de graves problèmes. Il peut par exemple offrir des performances insuffisantes, être difficile à entretenir et cher à faire fonctionner, 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 dans la figure ci-dessous décrit les étapes de planification, de conception et d'implémentation d'une solution logicielle d'entreprise reposant sur Java Enterprise System. Utilisez-le comme un outil permettant d'éviter toute dérive de votre projet de déploiement.

Figure 1-1 Cycle de vie de la solution



Le cycle de vie est constitué de phases ordonnées. Chaque phase se compose de tâches associées qui produisent des résultats utilisés comme base pour les phases suivantes. Les tâches constituant chaque phase sont itératives et exigent une analyse et une conception minutieuses pour que les résultats de la tâche en question puissent être générés. Les phases initiales peuvent également être itératives. Par exemple, au cours de la phase de conception du déploiement, vous pouvez vous rendre compte que l'analyse de la phase précédente n'est pas satisfaisante et qu'elle doit être réexaminée.

Les sections suivantes de ce chapitre fournissent une description succincte de chaque phase du cycle de vie.

Phase d'analyse d'exploitation

Au cours de cette phase, vous définissez les objectifs d'exploitation d'un projet de déploiement et formulez les exigences de l'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 réussite de la planification de votre déploiement et, finalement, celle de votre solution de déploiement en fonction des analyses effectuées au cours de la phase d'analyse d'exploitation.

Pendant la phase d'analyse d'exploitation, vous créez des documents d'exigences d'entreprise que vous utiliserez comme base pour la phase d'exigences techniques.

Pour plus d'informations sur la phase d'analyse d'exploitation, reportez-vous au [chapitre 2, « Analyse d'exploitation », page 27](#).

Phase d'exigences techniques

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 qui seront utilisées pour 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 d'exigences techniques, vous créez des documents contenant les informations suivantes :

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

Les documents d'analyse d'utilisation, de cas d'utilisation et d'exigences de qualité de service produits 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 d'exigences techniques, 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 et 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](#), « Exigences techniques », page 37.

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 identifiés lors de la phase d'exigences techniques. Vous identifiez également 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. En elle-même, celle-ci ne suffit pas pour passer à la phase de conception du déploiement. Les exigences de qualité de service établies lors de la phase d'exigences techniques sont également nécessaires. L'architecture logique et les exigences de qualité de service issues de la phase d'exigences techniques constituent un scénario de déploiement. Ce scénario sert de base à la phase de conception du déploiement.

Pour plus d'informations sur la conception logique, reportez-vous au [chapitre 4](#), « Conception logique », page 53.

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. Des plans et des spécifications détaillant différents aspects de l'implémentation de la solution logicielle sont également produits.

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, les contrats relatifs à l'implémentation du déploiement sont signés et les ressources destinées à la mise en œuvre du projet sont acquises. 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 réseau.
- **Spécifications d'implémentation** : spécifications détaillées utilisées comme base de travail pour le 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, « Conception du déploiement », page 75](#).

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 ou de l'intégralité des tâches suivantes :

- 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 passez 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, « Implémentation d'une conception de déploiement », page 109.](#)

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 pour le fonctionnement normal et non planifiées selon les besoins ;
- mise à niveau du logiciel et du matériel en cas de besoin.

Ce manuel n'a pas pour objet de décrire en détail la phase de fonctionnement.

Analyse d'exploitation

La phase d'analyse d'exploitation du cycle de vie de la solution 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 :

- « [A propos de l'analyse d'exploitation](#) », page 27
- « [Définition des exigences de l'entreprise](#) », page 28
- « [Définition de contraintes d'exploitation](#) », page 33

A 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 entraîne la création de documents d'exigences d'exploitation.

Vous utiliserez ces documents au cours de la phase d'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 grâce à un effort de réflexion créative.

Cette section décrit 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 définit 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.

Portée

Les exigences de l'entreprise doivent préciser la portée du projet de déploiement. Veillez à identifier les problèmes pouvant être résolus et évitez de définir des exigences « ouvertes » qui nuisent à la compréhension et à la réalisation des objectifs. Une portée mal définie peut entraîner des problèmes de déploiement (non adapté aux besoins de l'entreprise ou trop exigeant en ressources).

Priorités

Affectez des priorités à vos objectifs 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, l'ampleur et la complexité de certains déploiements exigent généralement une implémentation progressive de la solution. La définition de priorités fournit une aide à la prise des décisions qui détermineront l'acceptation de votre projet de déploiement par les parties prenantes.

Qualités essentielles

Identifiez les secteurs clés de la réussite 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 changer 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 événements externes imprévus). Veillez à ménager 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 savoir à quels types d'utilisateurs la solution s'adresse, quels sont leurs besoins et quels bénéfices ils en attendent. Par exemple, pour classer les utilisateurs, vous pouvez utiliser les catégories suivantes :

- employés actuels uniquement ;
- employés actuels et anciens employés ;
- administrateurs ;
- clients actifs ;
- tous les clients ;
- site d'appartenance ;
- public général ;
- accès restreint.

La définition des attentes des utilisateurs permet 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 ;
- autogestion 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 durée de disponibilité ;
- réduction des taux d'erreurs ;
- archivage et conservation des informations.

Exprimez les exigences opérationnelles dans des termes clairs accessibles à toutes les parties prenantes. Évitez les termes ambigus tels que « temps de réponse adéquats ». Voici quelques exemples d'exigences opérationnelles :

- reprise du fonctionnement dans les 10 minutes suivant une panne ;
- possibilité de restituer les messages reçus au cours des dernières 48 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 modèles d'utilisation existants

Exprimez les modèles d'utilisation existants en tant qu'objectifs mesurables. Pour déterminer ces objectifs, vous pouvez vous poser les questions suivantes :

- Comment les services actuels sont-ils utilisés ?
- Quels sont les modèles d'utilisation (occasionnelle, fréquente ou intense) ?
- À quels sites les utilisateurs se connectent-ils 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 la fréquence et la durée d'accès des utilisateurs aux services existants. Si votre expérience ne vous permet pas d'établir ces modèles d'utilisation, étudiez ceux en vigueur dans des entreprises similaires.

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 individus et les organisations qui sont 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, les parties prenantes doivent également être informées des modifications planifiées. Une partie prenante insuffisamment impliquée ou mal informée pourrait bloquer l'implémentation du déploiement.

Normes et stratégies

Une bonne compréhension des normes et des stratégies de l'entreprise demandeuse de la solution est essentielle. 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 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 la nature de l'entreprise. 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, prenez en compte les lois et les réglementations en vigueur dans les pays étrangers concernés. Par exemple, la plupart des pays européens exercent un contrôle strict sur le stockage des informations personnelles.

Sécurité

Certains de vos objectifs peuvent soulever des problèmes de sécurité qui doivent être identifiés. Mettez en évidence les objectifs de sécurité liés à la réussite de la solution. Par exemple :

- accès autorisé aux informations propriétaires ;
- accès basé sur des rôles aux 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 règles de sécurité.

Répartition géographique

La répartition géographique des sites et la largeur de bande entre ces derniers peuvent avoir une incidence sur les décisions de conception. Certains sites peuvent également 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 la réussite de la solution.

Approche incrémentielle

Les solutions logicielles sont généralement considérées comme des systèmes complets. Toutefois, leur déploiement s'effectue généralement 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 ;
- mobilisation moindre des ressources humaines ;
- possibilité d'envisager des partenariats.

Contrats de niveau de service (SLA)

Un contrat de niveau de service (SLA, Service Level Agreement) 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. 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 d'exigences techniques. 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 après incident. 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 la portée d'un SLA, vous devez définir les attentes de l'entreprise en matière de disponibilité du système.

Définition de contraintes d'exploitation

Ces contraintes jouent un rôle important dans la définition de la nature d'un projet de déploiement. Un déploiement réussi repose sur la réalisation des objectifs d'exploitation dans le respect des contraintes d'exploitation. Celles-ci peuvent être fiscales, physiques (la capacité du réseau, par exemple), temporelles (fin du projet avant un événement important tel que la prochaine réunion annuelle, par exemple) ou toute autre limitation susceptible d'affecter la réalisation des objectifs d'exploitation.

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

Problèmes de migration

Généralement, 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 du planning

Le planning d'implémentation d'une solution peut avoir des répercussions sur les décisions de conception. Un planning trop agressif peut entraîner une révision à la baisse des objectifs, une modification des priorités ou l'application d'une approche incrémentielle. Certaines étapes importantes d'un planning peuvent également nécessiter une attention particulière. Ces étapes peuvent être déterminées par des événements internes (changements de service planifiés, par exemple) ou externes (date de début d'un trimestre scolaire, par exemple).

Limitations budgétaires

La plupart des projets de déploiement doivent être menés dans le cadre d'un 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 se fonde 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** : des ressources limitées (matérielles, logicielles et humaines) peuvent suggérer un déploiement incrémentiel. Vous devrez peut-être faire appel aux mêmes ressources ou équipes de développement au cours de chaque phase incrémentielle implémentée.
- **Maintenance, administration et support** : analysez les ressources disponibles pour ces trois fonctions. Des ressources limitées peuvent avoir un impact sur les décisions de conception.

Coût de propriété

Outre la maintenance, l'administration et le support, analysez les autres facteurs qui interviennent dans le coût de propriété. Les mises à niveau matérielles et logicielles requises, l'impact de la solution sur le réseau électrique, le coût des télécommunications et d'autres facteurs augmentent les frais courants. Les SLA spécifiant des niveaux de disponibilité affectent également le coût de propriété car ils exigent une redondance accrue.

L'implémentation d'une solution doit trouver un retour sur investissement dans la solution même. L'analyse du retour sur investissement consiste à estimer les bénéfices engendrés par rapport aux capitaux dépensés.

Pour évaluer les avantages d'une solution, vous devez analyser attentivement les objectifs visés et effectuer une comparaison avec d'autres solutions permettant de les atteindre. Si vous choisissez de ne rien faire, vous devez également évaluer le coût de cette solution.

Exigences techniques

La phase d'exigences techniques du cycle de vie de la solution 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 », page 37
- « Analyse d'utilisation », page 38
- « Cas d'utilisation », page 40
- « Exigences de qualité de service », page 41
- « Exigences de niveau de service », page 51

À propos des exigences techniques

L'analyse des exigences techniques débute par l'examen des exigences de l'entreprise stipulées lors de la phase d'analyse d'exploitation du cycle de vie de la solution. 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 qui façonnent une interaction utilisateur normale 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 dans la phase de conception du déploiement du cycle de vie de la solution.

Comme pour 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 à venir ainsi que la définition de leurs modèles 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'attribution de pondérations aux cas d'utilisation, comme décrit dans la section « [Cas d'utilisation](#) », page 40.

Au cours de l'analyse d'utilisation, vous devez consulter les utilisateurs dès que vous en avez la possibilité, rechercher les données sur les modèles d'utilisation existants et interroger les concepteurs et administrateurs des systèmes précédents. Le tableau suivant répertorie les paramètres dont il faut tenir compte lors d'une analyse d'utilisation.

Tableau 3-1 Paramètres à prendre en considération lors de l'analyse d'utilisation

Paramètre	Description
Nombre et type d'utilisateurs	<p>Identifiez le nombre d'utilisateurs que la solution doit prendre en charge et classez-les, si nécessaire.</p> <p>Par exemple :</p> <ul style="list-style-type: none"> • Une solution B2C (Business to Customer, transactions entre une entreprise et des particuliers) peut comporter un grand nombre de visiteurs, mais seul un petit nombre d'entre eux s'enregistrent et s'engagent dans des transactions d'entreprises. • Une solution B2E (Business to Employee, transactions entre une entreprise et ses salariés) doit prendre en compte chaque employé, sachant que certains employés peuvent avoir besoin d'accéder au réseau interne à partir de l'extérieur. <p>Dans ce type de solution, il se peut que les dirigeants aient besoin d'une autorisation pour pouvoir accéder à certaines pages inaccessibles aux employés normaux.</p>
Utilisateurs actifs et inactifs	<p>Identifiez les modèles 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 accèdent au système déployé pour contrôler, mettre à jour et prendre en charge le déploiement.</p> <p>Déterminez les modèles d'utilisation administratifs spécifiques pouvant affecter les exigences techniques (par exemple, l'administration du déploiement en dehors du pare-feu).</p>
Modèles d'utilisation	<p>Identifiez la façon dont les différents types d'utilisateurs accèdent au système et fournissent des objectifs pour l'utilisation attendue.</p> <p>Par exemple :</p> <ul style="list-style-type: none"> • Existe-t-il des périodes d'utilisation de pointe ? • Quelles sont les heures de bureau normales ? • Les utilisateurs sont-ils répartis de façon équitable ? • Quelle est la durée moyenne d'une connexion utilisateur ?

Tableau 3-1 Paramètres à prendre en considération lors de l'analyse d'utilisation (*Suite*)

Paramètre	Description
Augmentation du nombre d'utilisateurs	Déterminez si le nombre d'utilisateurs est fixe ou si le déploiement implique une augmentation du nombre d'utilisateurs. Si une augmentation est attendue, tentez de créer des prévisions raisonnables de cet accroissement.
Transactions utilisateur	Identifiez le type de transactions utilisateur à prendre en charge. Ces transactions peuvent être converties en cas d'utilisation. Par exemple : <ul style="list-style-type: none"> • Quelles sont les tâches effectuées par les utilisateurs ? • Une fois l'utilisateur connecté, la connexion est-elle maintenue ? Généralement, effectuent-ils quelques tâches avant de se déconnecter ? • Une collaboration importante entre utilisateurs requiert-elle des calendriers communs, des conférences Web et le déploiement de pages Web internes ?
Études d'utilisateurs et données statistiques	Utilisez les études d'utilisateurs et les autres sources qui existent déjà pour déterminer les modèles du comportement de l'utilisateur. Souvent, les entreprises ou les organisations industrielles effectuent des études sur la recherche d'utilisateurs à partir desquelles il est possible d'extraire des informations utiles sur les utilisateurs. Les fichiers journaux des applications existantes peuvent contenir des données statistiques utiles pour les estimations d'un système.

Cas d'utilisation

Les cas d'utilisation façonnent une interaction utilisateur classique avec la solution que vous concevez et décrivent le déroulement complet d'une opération du point de vue d'un utilisateur final. Le classement par ordre de priorité de la conception autour d'un ensemble complet de cas d'utilisation assure la livraison des fonctionnalités prévues. Les cas d'utilisation représentent les entrées principales de la conception logique.

Attribuez des pondérations relatives aux cas d'utilisation, sachant que plus la pondération est élevée et plus les tâches utilisateur sont 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 divisés en deux niveaux.

- **Rapports sur les cas d'utilisation** : descriptions des cas d'utilisation individuels, y compris les flux d'événements principaux et alternatifs.
- **Diagrammes des cas d'utilisation** : diagrammes illustrant les relations entre les acteurs et les cas d'utilisation, présentant une organisation plus formelle du flux des événements. Les diagrammes des cas d'utilisation permettent de créer des cas d'utilisation étendus ou complexes. En règle générale, la norme UML (Unified Modeling Language) est utilisée pour dessiner les diagrammes relatifs aux cas d'utilisation.

Exigences de qualité de service

Les exigences de qualité de service sont des spécifications techniques indiquant les qualités système des fonctions, telles que les performances, la disponibilité, l'évolutivité et l'entretien. Ces exigences 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 satisfaire cette exigence de l'entreprise.

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

Tableau 3-2 Qualités systèmes 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 connue sous le nom de <i>temps d'activité</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 elle ne doit pas requérir 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 implique l'authentification et l'autorisation des utilisateurs, la sécurité des données ainsi que l'accès sécurisé à un système déployé.

Tableau 3-2 Qualités systèmes ayant un impact sur les exigences de qualité de service

Qualité du système	Description
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 comprend le contrôle du système, la réparation des problèmes se produisant et la mise à niveau des composants matériels et logiciels.

Les qualités système sont étroitement liées. Les exigences liées à un critère de qualité système peuvent affecter 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'avoir une influence sur la disponibilité. L'ajout de serveurs supplémentaires en vue de pallier les problèmes de disponibilité peut affecter l'entretien (coûts de maintenance).

Il est capital de comprendre la manière dont les qualités système sont liées et de savoir quels compromis sont à faire pour concevoir un système qui satisfait les exigences et les contraintes d'entreprise.

Les sections suivantes décrivent, de façon plus détaillée, les qualités système ayant un impact sur la conception de déploiement et fournissent des informations sur les paramètres dont il faut tenir compte lors de l'élaboration des 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 indiquant une exigence d'entreprise relatif à l'accès Web :

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

En commençant par cette exigence d'entreprise, 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 terme 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 stipulant les exigences système en termes de performances :

Le temps de réponse lié à l'actualisation des pages Web, échantillonné toutes les 15 minutes, ne doit pas dépasser quatre secondes pendant la journée 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 et le nombre d'erreurs par million de transactions ne doit 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 pics de charge inhabituels).

Disponibilité

La disponibilité permet de définir le temps d'activité du système. Celle-ci indique généralement le pourcentage de temps pendant lequel le système est accessible aux utilisateurs. Une interruption d'activité, 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 tout autre paramètre (par exemple, une coupure d'alimentation) engendrant une panne système. Une interruption planifiée de l'exploitation en vue d'effectuer des opérations d'entretien (maintenance et mises à niveau) n'est pas considérée comme une interruption d'activité en tant que telle. Voici une équation élémentaire permettant de calculer la disponibilité du système en terme de pourcentage du temps d'activité :

$$\text{Disponibilité} = \frac{\text{temps de disponibilité}}{(\text{temps de disponibilité} + \text{interruption d'activité})} \times 100 \%$$

En règle générale, la disponibilité est fonction du nombre de « neuf » obtenu. Par exemple, une disponibilité de 99 % comporte deux neuf. Lorsque vous indiquez davantage de neuf, la conception du déploiement est fortement modifiée. Le tableau suivant indique le temps d'interruption d'activité non planifiée, 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	Interruption d'activité 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 comportant 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 (tels que les processeurs, la mémoire et les périphériques réseau) ou logicielle permettant d'assurer les services essentiels.

Un point de panne unique est 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.

Les systèmes à tolérance de panne peuvent s'avérer coûteux à implémenter et 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 de la disponibilité du service

D'un point de vue utilisateur, la disponibilité prend davantage en compte les services individuels 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 plus 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	De haute importance	Il s'agit des services qui doivent être disponibles en permanence. Par exemple, les services de bases de données (tels que les annuaires LDAP) aux applications.
2	Doit être disponible	Il s'agit des services qui doivent rester disponibles, mais pour lesquels les performances peuvent être réduites. Par exemple, il se peut que la disponibilité du service de messagerie ne soit pas critique dans certains environnements de travail.
3	Peut être différé	Il s'agit des services qui doivent être disponibles dans un temps donné. Par exemple, il se peut que la disponibilité des services de calendrier ne soit pas essentielles dans certains environnements de travail.
4	Facultatif	Il s'agit des services qui peuvent être différés indéfiniment. 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 réagit face à ces situations.

Évolutivité

L'évolutivité correspond à la possibilité d'ajouter de la capacité à un système de sorte qu'il puisse prendre en charge une charge supplémentaire provenant d'utilisateurs existants ou d'un nombre plus élevé d'utilisateurs. L'évolutivité requiert généralement davantage de ressources mais ne doit pas impliquer de modification au niveau de la conception de l'architecture du déploiement ni de perte de service liée au temps nécessaire à l'ajout des ressources supplémentaires.

Comme pour la disponibilité, l'évolutivité prend davantage en compte les services individuels fournis par un système que la totalité du système. En revanche, pour les services dont dépendent d'autres services, tels que 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 la croissance du déploiement est clairement définie dans les exigences de l'entreprise. Cependant, lors de la phase de conception de déploiement du cycle de vie de la solution, l'architecture de déploiement doit toujours prévoir une certaine tolérance pour la mise à l'échelle le système et ce, même si aucune exigence de qualité de service concernant l'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, d'estimations et de spéculations aléatoires. Vous devez tenir compte des trois éléments ci-dessous pour élaborer les 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 des jalons précis de la mise à l'échelle du système. On appelle jalons les exigences de charge fixées à l'avance pour l'évaluation de 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 de performances peuvent fournir une aide aux opérateurs et administrateurs responsables de la maintenance et des mises à niveau.

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

Tableau 3-5 Facteurs d'évolutivité

Paramètre	Description
Analyse de modèles d'utilisation	En examinant les données existantes, vous pouvez comprendre les modèles d'utilisation des utilisateurs actuels ou prévus. En l'absence de telles données, analysez les données du secteur ou les estimations du marché.
Conception pour une échelle maximale acceptable	<p>Conception ayant pour objectif l'échelle maximale requise pour satisfaire les demandes connues et celles possibles.</p> <p>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 attentes acceptables en matière de charge à venir. La durée de l'estimation dépend fortement de la fiabilité des projections.</p>
Définition de jalons appropriés	<p>Implémentez la conception du déploiement par incrément afin de répondre aux exigences à court terme en utilisant une mémoire pour faire face à une croissance inattendue. Définissez des jalons pour l'ajout de ressources système.</p> <p>Par exemple :</p> <ul style="list-style-type: none"> • Achat essentiel (trimestriel ou annuel) • Délai d'exécution pour l'achat 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 d'appréhender les nouvelles technologies, telles que les processeurs et les serveurs Web plus rapides, ainsi que la façon dont elles affectent les performances de l'architecture sous-jacente.

Sécurité

La sécurité est un sujet complexe qui implique 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 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 réduire 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 recrute un architecte de sécurité chargé de la conception et de l'implémentation des mesures de sécurité.

La section suivante décrit quelques-uns des domaines abordés dans la 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 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 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-feu, 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 pour 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 assurer 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 est 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 paramètre important pour la disponibilité, les performances et l'évolutivité du système.

Exigences d'entretien

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

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

Tableau 3-6 Paramètres concernant les exigences d'entretien

Paramètre	Description
Planification de l'interruption d'activité	<p>Identification des tâches de maintenance nécessitant que des services spécifiques soient non disponibles ou partiellement non disponibles.</p> <p>Certaines opérations de maintenance et de mise à niveau peuvent être invisibles pour l'utilisateur, tandis que d'autres nécessitent une interruption des services. Lorsque cela est possible, planifiez avec les utilisateurs les activités de maintenance nécessitant une interruption d'activité afin qu'ils puissent prévoir ces arrêts.</p>

Tableau 3-6 Paramètres concernant les exigences d'entretien (*Suite*)

Paramètre	Description
Modèles d'utilisation	<p>Identification des modèles d'utilisation en vue déterminer le meilleur moment pour la planification des opérations de maintenance.</p> <p>Par exemple, programmez les opérations de maintenance en soirée ou en week-end sur les systèmes sur lesquels les pics d'utilisation se produisent pendant les heures de bureau normales. Pour les systèmes distribués géographiquement, l'identification de ces heures peut s'avérer plus difficile.</p>
Disponibilité	<p>L'entretien reflète généralement votre conception de disponibilité. Les stratégies de réduction de l'interruption d'activité pour la maintenance et les mises à niveau se rapportent à votre stratégie de disponibilité. Les systèmes nécessitant un degré de disponibilité élevé ne disposent que de peu de temps pour effectuer la maintenance, les mises à niveau et les réparations.</p> <p>Les stratégies de gestion des exigences de disponibilité affectent 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 nécessiter des 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 des zones problématiques.</p> <p>Le contrôle régulier d'un système permet d'éviter des problèmes, de gérer des charges de travail selon les stratégies de disponibilité et d'améliorer la planification de la maintenance et de l'interruption d'activité.</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 représentent une garantie concernant la qualité globale du système que le système déployé doit satisfaire. 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.

Exigences de niveau de service

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 d'exigences techniques, 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 », page 53
- « Conception d'une architecture logique », page 55
- « Exemples d'architectures logiques », page 63
- « Zones d'accès », page 72
- « Scénario de déploiement », page 73

À propos des architectures logiques

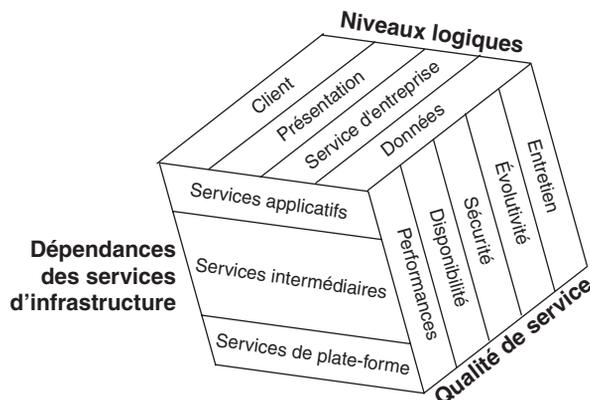
L'architecture logique identifie les composants logiciels nécessaires à l'implémentation d'une solution et décrit les relations existant entre ces composants. L'architecture logique et les exigences de qualité de service définies lors de la phase d'exigences techniques constituent un scénario de déploiement. Celui-ci sert de base à la conception de l'architecture de déploiement, qui intervient au cours de la phase suivante, 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 [Figure 4-1, page 54](#).

- **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 en niveaux qui représentent l'indépendance logique et physique des composants logiciels, selon la nature des services qu'ils fournissent.
- **Qualité de service** : qualités de service système, telles que les performances, la disponibilité, l'évolutivité et d'autres qualités qui illustrent divers aspects de la conception et du fonctionnement d'une solution logicielle.

Figure 4-1 Trois dimensions de l'architecture de la solution Java Enterprise System



REMARQUE Pour plus d'informations sur les concepts d'architecture de Java Enterprise System, consultez le chapitre « Architectures de la solution Java Enterprise System » du manuel *Présentation technique de Sun Java Enterprise System 2005T1*, à l'adresse <http://docs.sun.com/doc/819-1925>.

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 aux niveaux clients. 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 d'exigences techniques 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 identifiés initialement.

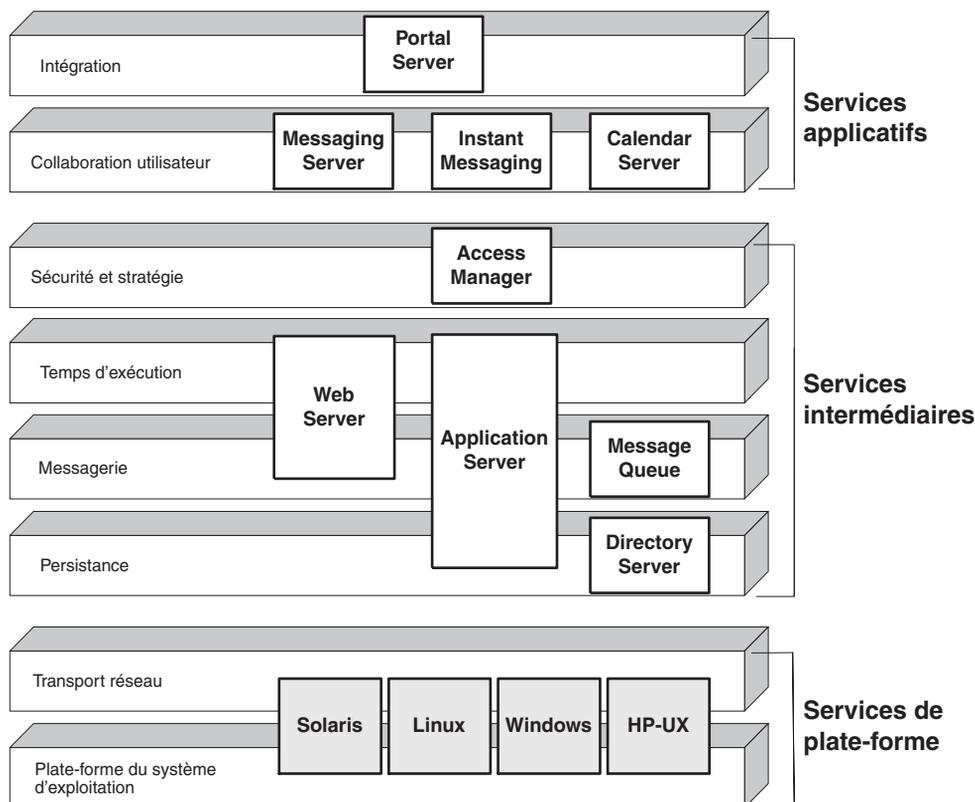
Les composants de Java Enterprise System sont positionnés dans une architecture à plusieurs niveaux en fonction du type de service qu'ils fournissent. Cette approche vous permettra de définir la distribution des services fournis par les composants et d'élaborer une stratégie d'implémentation des exigences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

Vous pouvez également répartir les composants logiques dans des zones d'accès sécurisé. La section « [Zones d'accès](#) », page 72, propose des exemples de ces zones.

Composants de Java Enterprise System

Java Enterprise System est constitué de composants logiciels en interaction qui fournissent des services d'entreprise que vous pouvez utiliser pour créer votre solution. La figure ci-dessous illustre les principaux composants logiciels de Java Enterprise System. Le manuel *Présentation technique de Sun Java Enterprise System 2005T1*, <http://docs.sun.com/doc/819-1925>, contient des informations supplémentaires sur les composants de Java Enterprise System et les services qu'ils fournissent.

Figure 4-2 Composants de 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 comme composant requis d'une architecture logique, celle-ci doit également comprendre Directory Server et, éventuellement, Access Manager. En effet, Messaging Server dépend de Directory Server pour les services d'annuaire et de Access Manager pour les solutions exigeant une connexion unique.

Le tableau ci-dessous répertorie les dépendances entre les composants de Java Enterprise System. La [Figure 4-3, page 59](#) propose une représentation visuelle des dépendances entre les composants clés. Lors de la conception d'une architecture logique, utilisez ce tableau et la figure associée pour identifier les composants dépendants de votre conception.

Tableau 4-1 Java Enterprise System Dépendances entre composants de

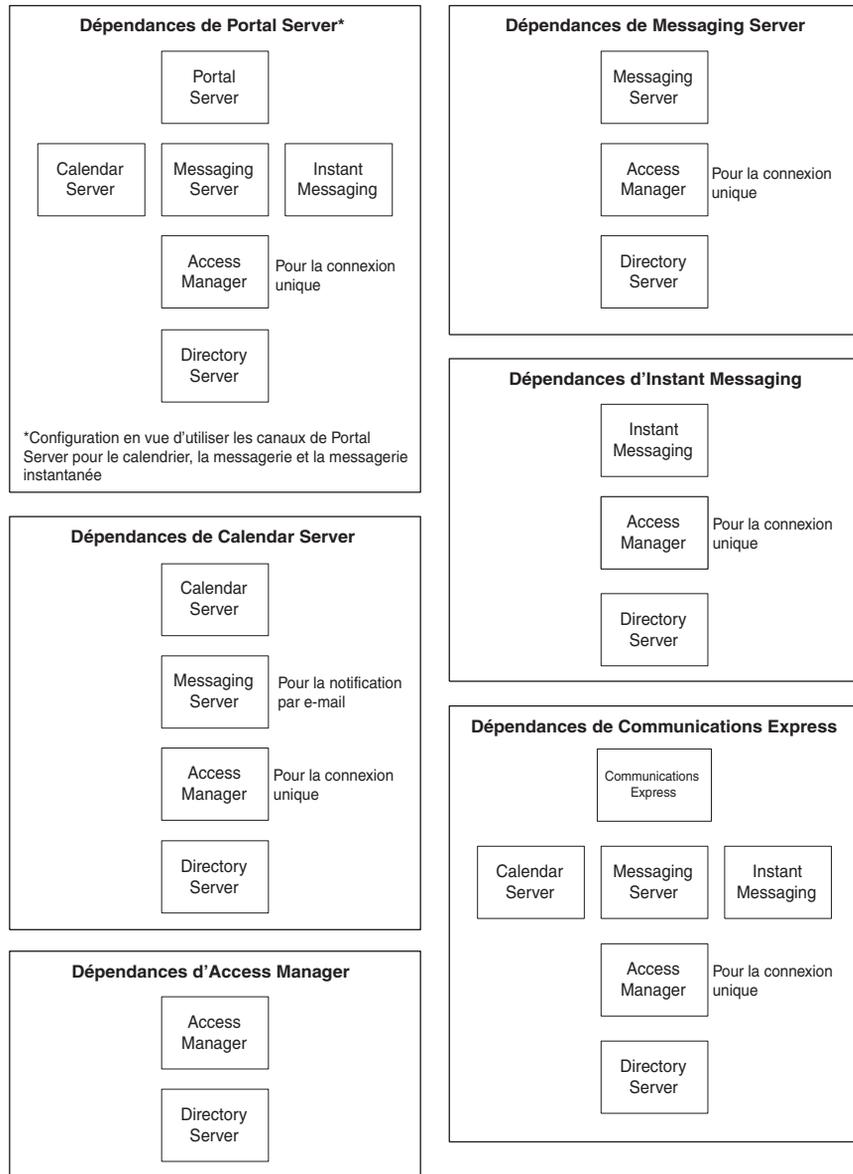
Composant de Java Enterprise System	Dépend de
Application Server	Message Queue Directory Server (facultatif)
Calendar Server	Messaging Server (pour le service de notification par e-mail) Access Manager (pour la connexion unique) Web Server (pour l'interface Web) Directory Server
Communications Express	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	Rien
Access Manager	Application Server ou Web Server Directory Server
Instant Messaging	Access Manager (pour la connexion unique) Directory Server

Tableau 4-1 Java Enterprise System Dépendances entre composants de *(Suite)*

Composant de Java Enterprise System	Dépend de
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 répertoriées dans le [Tableau 4-1](#) n'est pas exhaustive. Le [Tableau 4-1](#) ne présente pas les dépendances que vous devez prendre en compte lors de la planification de l'installation. Pour une liste complète des dépendances Java Enterprise System, reportez-vous au *Guide d'installation de Sun Java Enterprise System 2005T1*, <http://docs.sun.com/doc/819-0807>.

Figure 4-3 Dépendances entre composants de Java Enterprise System



Support de conteneur Web

La section précédente, « [Dépendances entre composants de](#) », ne mentionne pas le 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 prendre en compte le conteneur Web nécessaire à ces composants.

Services logiques distincts fournis par Messaging Server

Java Enterprise System Messaging Server 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 représentées sur 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 « [Exemples d'architectures logiques](#) », [page 63](#), propose des exemples de composants logiques distincts.

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

Tableau 4-2 Messaging Server Configurations

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 en interne (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, fréquemment depuis des sites situés hors du pare-feu de l'entreprise. Certaines configurations de Messaging Server peuvent également fournir un accès réseau (Messaging Server 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

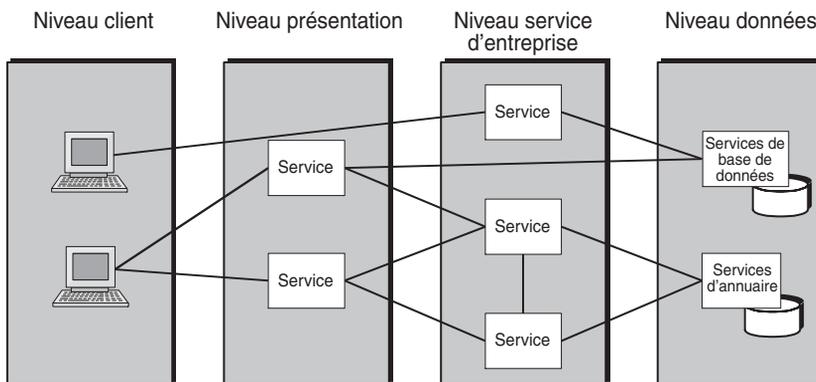
Composant	Description
Directory Proxy Server	Fournit des services améliorés d'accès aux annuaires, de compatibilité de schéma, de routage et d'équilibrage de charge pour des instances multiples de Directory Server.
Portal Server, Portal Server Secure Remote Access	Fournit un accès Internet sécurisé depuis l'extérieur d'un pare-feu d'entreprise au contenu et services de Portal Server, y compris les portails internes et les applications Internet.
Portal Server, Portal Server Mobile Access	Fournit un accès sans fil depuis des périphériques mobiles et un accès vocal à Portal Server.
Messaging ServerMessage 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](#) », page 72.

Conception d'architecture à plusieurs niveaux

Java Enterprise System est adapté à la conception d'architectures à plusieurs niveaux, dans lesquelles les services sont répartis sur différents niveaux 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. La figure ci-dessous décrit un modèle d'architecture à plusieurs niveaux pour des applications d'entreprise, avec les niveaux client, présentation, services d'entreprise et données.

Figure 4-4 Modèle d'architecture à plusieurs niveaux



Le tableau ci-dessous présente les niveaux logiques décrits dans la [Figure 4-4](#).

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 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 généralement chargés de récupérer les données du niveau données 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 l'accès à l'annuaire LDAP aux 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 de les distribuer au sein de votre réseau. Vous pouvez également voir comment les composants de l'architecture accèdent aux services des autres composants. Cette visibilité vous permet de planifier les solutions de qualité de service en termes de disponibilité, d'évolutivité, de sécurité, etc.

Exemples d'architectures logiques

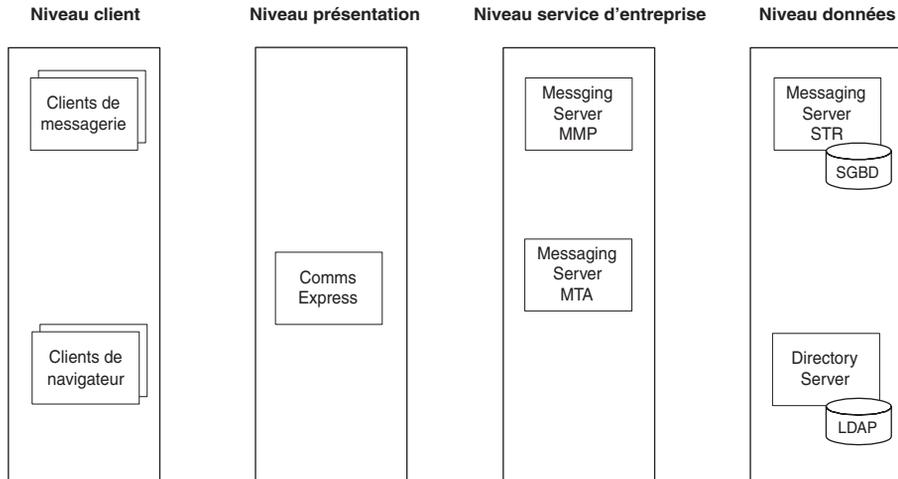
Cette section présente des exemples d'architectures logiques pour les solutions Java Enterprise System. Ces exemples illustrent la répartition des composants logiques sur les niveaux appropriés de l'architecture à plusieurs niveaux, puis l'analyse des relations entre composants via l'étude des cas d'utilisation. 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 représente une solution Messaging Server de base qui illustre 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 employant de 1 000 à 5 000 personnes.

Exemple Messaging Server

La figure 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 décrites dans les figures ci-après.

REMARQUE Généralement, un déploiement de Messaging Server fait partie d'une solution d'entreprise incluant d'autres composants de Java Enterprise System, comme l'explique la section « [Exemple de communications basées sur les identités](#) », page 68.

Figure 4-5 Architecture logique de déploiement de Messaging Server

Le tableau ci-dessous décrit les composants présentés dans la [Figure 4-5](#).

Tableau 4-5 Composants d'une architecture logique de Messaging Server

Composant	Description
Clients de messagerie	Applications clientes pour la lecture et l'envoi de messages.
Messaging Server MTA	Messaging Server configuré en tant qu'agent de transfert de messages (MTA) pour recevoir, acheminer, transporter et livrer des messages.
Messaging Server MMP	Messaging Server configuré 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	Messaging Server configuré 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. Par exemple, dans les déploiements d'entreprise, des instances de MTA entrant et sortant séparées sont généralement créées, mais la [Figure 4-5, page 64](#) ne montre qu'un composant MTA. La décision de répliquer les composants logiques en plusieurs instances est prise lors de phase de conception du déploiement.

Cas d'utilisation de Messaging Server

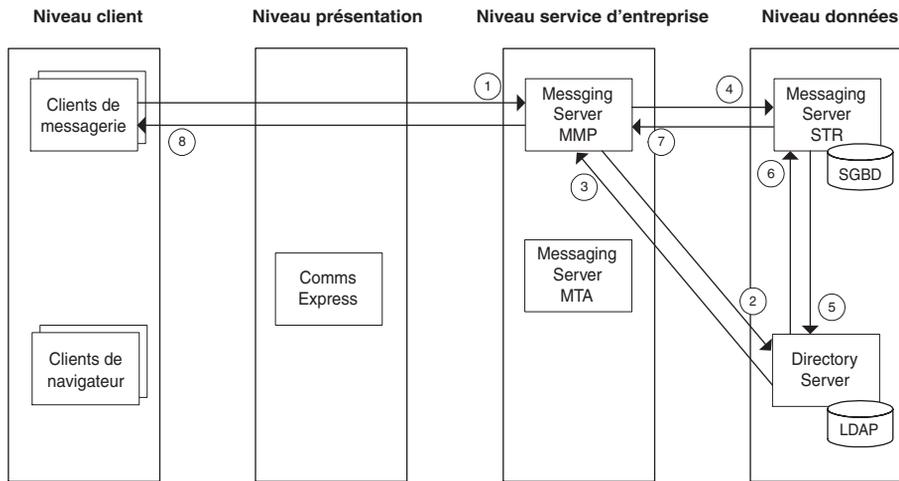
Les cas d'utilisation permettent d'identifier les relations entre les composants logiques d'une architecture. En mappant les interactions entre composants en fonction des cas d'utilisation, vous obtenez une représentation visuelle de ces interactions qui vous sera utile pour la conception du déploiement.

L'analyse des cas d'utilisation permet généralement de déterminer les interactions entre composants avant de commencer la conception du déploiement. Les trois cas d'utilisation ci-dessous, propres à Messaging Server, illustrent les interactions entre composants logiques.

Premier cas d'utilisation : la connexion de l'utilisateur à Messaging Server aboutit

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 message à MMP.
8. MMP achemine la liste de message vers le client de messagerie.

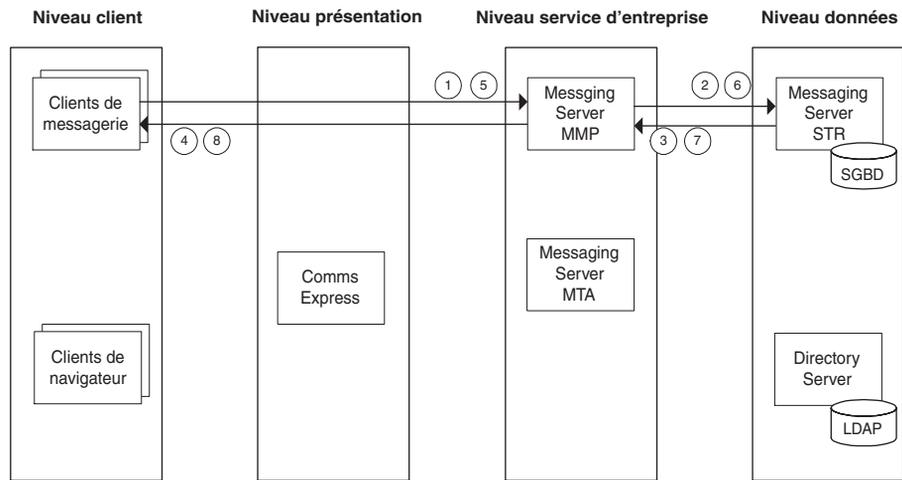
Figure 4-6 Architecture logique de Messaging Server illustrant le premier cas d'utilisation



Deuxième cas d'utilisation : l'utilisateur connecté lit et supprime des messages

1. Le client de messagerie demande le message à lire à Messaging Server Multiplexor (MMP).
2. MMP demande le message à Messaging Server Message Store (STR).
3. STR renvoie le message à MMP.
4. MMP achemine le message vers le client de messagerie.
5. Le client de messagerie envoie l'action de suppression de message à MMP.
6. MMP achemine l'action de suppression de message vers STR.
7. STR supprime le message de la base de données et envoie une confirmation à MMP.
8. MMP achemine la confirmation de suppression vers le client de messagerie.

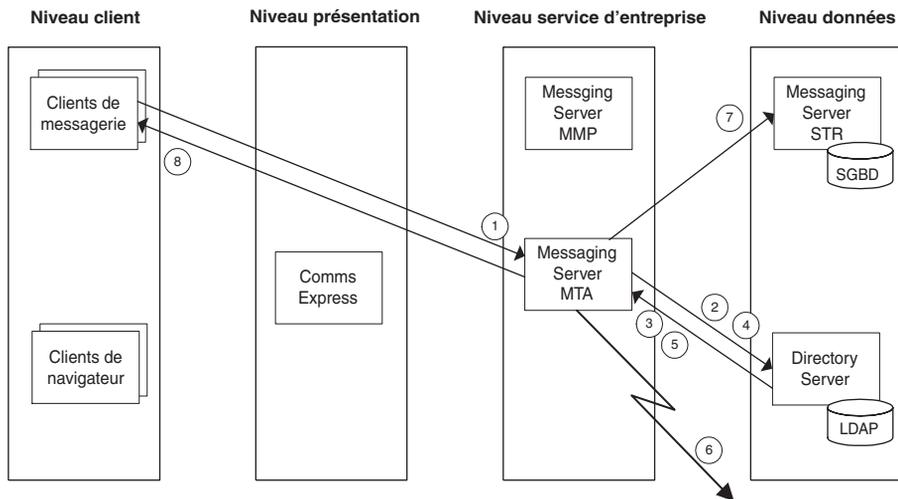
Figure 4-7 Architecture logique de Messaging Server illustrant le deuxième cas d'utilisation



Troisième cas d'utilisation : l'utilisateur connecté envoie un message

1. Le client de messagerie envoie un message composé dans le cadre du client 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 à 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 à MTA.
6. MTA achemine le message vers 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.

Figure 4-8 Architecture logique de Messaging Server illustrant le troisième cas d'utilisation



Exemple de communications basées sur les identités

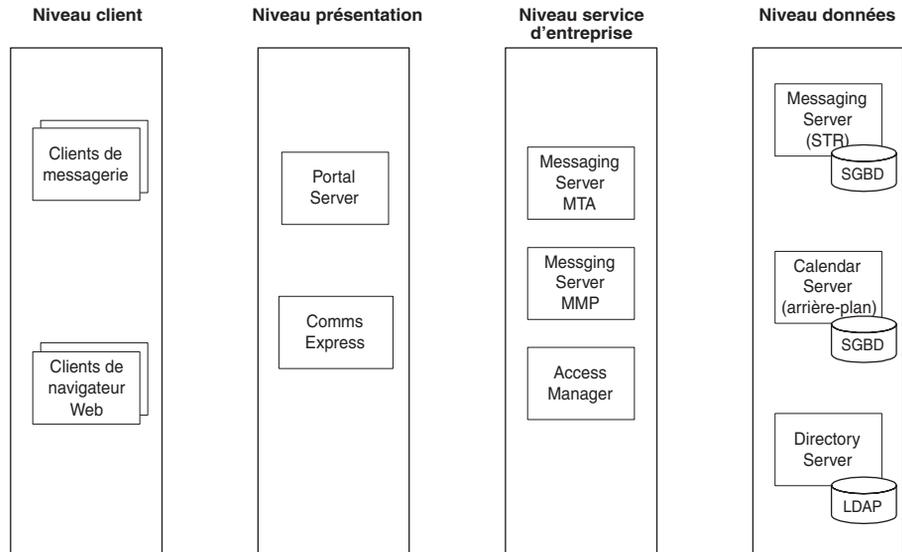
Cet exemple illustre une solution de communications basées sur les identités pour une entreprise de taille moyenne employant de 1 000 à 5 000 personnes. Généralement, une analyse d'exploitation exhaustive suivie d'une analyse détaillée des exigences techniques sont requises pour concevoir l'architecture logique. Toutefois, cet exemple étant théorique, il est permis de supposer 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 communications, aux services d'agenda et à d'autres ressources.
- L'authentification et l'autorisation à l'échelle de l'entreprise fournissent l'accès aux sites Web internes et à d'autres services.
- Les identités uniques sont suivies dans tous les services de l'entreprise, ce qui permet une connexion unique (Single Sign-On, 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 agendas et d'autres activités utilisateur du même type.

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

Figure 4-9 Architecture logique d'un scénario 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 de navigateur Web. Il divise ce scénario de connexion en deux cas d'utilisation :

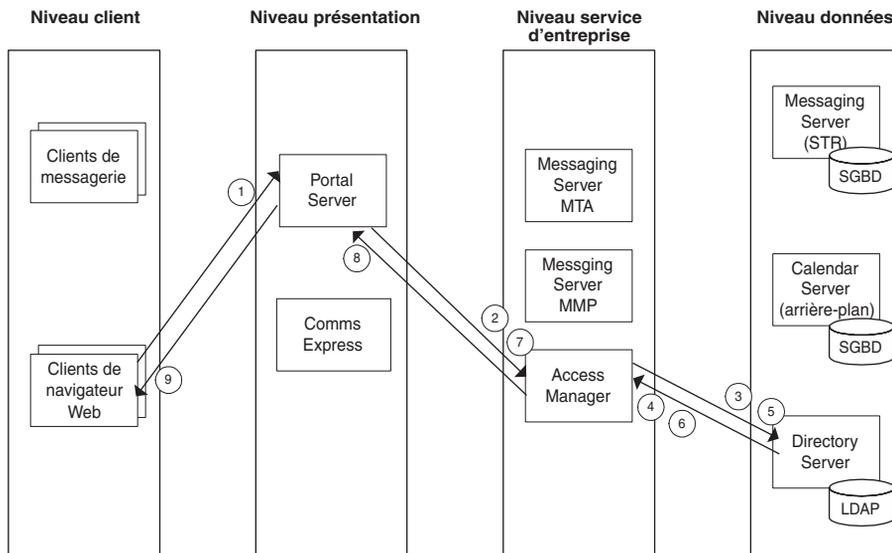
- L'utilisateur se connecte, est authentifié, puis Portal Server récupère sa configuration de portail.
- Portal Server récupère les informations de messagerie et d'agenda afin de les afficher dans le client Web.

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

Premier cas d'utilisation : la connexion de l'utilisateur aboutit et le portail récupère sa configuration

1. Le client de navigateur Web envoie l'ID utilisateur et le mot de passe à Portal Server.
2. Portal Server demande l'authentification à 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 est affichée dans le client de navigateur Web.

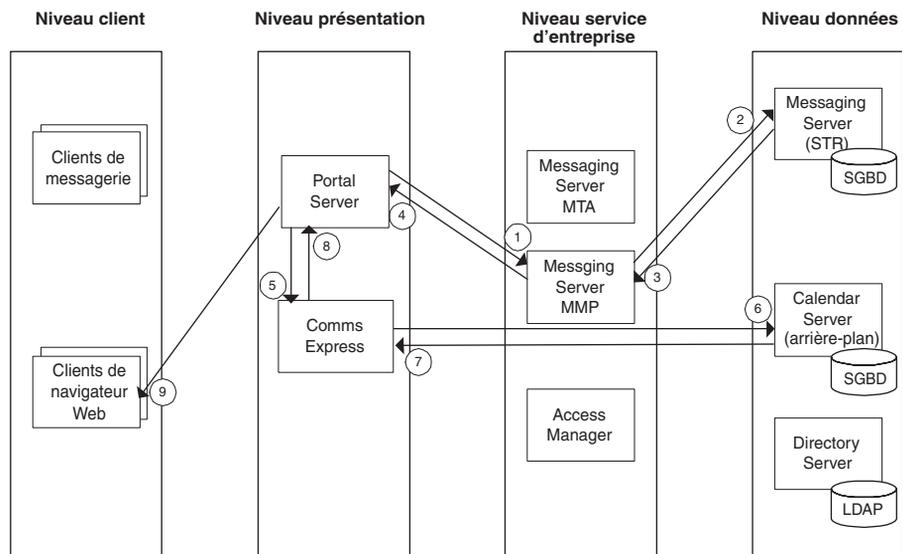
Figure 4-10 Architecture logique de scénario de communications illustrant le premier cas d'utilisation



Deuxième cas d'utilisation : Portal Server affiche les informations de messagerie et d'agenda

1. Une fois la connexion, l'authentification et la récupération de la configuration du portail réussies, Portal Server demande les messages à Messaging Server MMP.
2. MMP demande la liste de messages à Messaging Server STR.
3. STR renvoie la liste de message à MMP.
4. MMP achemine les en-têtes de messages vers Portal Server.
5. Portal Server demande les informations d'agenda à Communications Express.
6. Communications Express demande les informations d'agenda au composant d'arrière-plan de Calendar Server.
7. Le composant d'arrière-plan de Calendar Server renvoie les informations d'agenda à Communications Express.
8. Communications Express achemine les informations d'agenda vers Portal Server.
9. Portal Server envoie les informations de canal au client de navigateur Web.

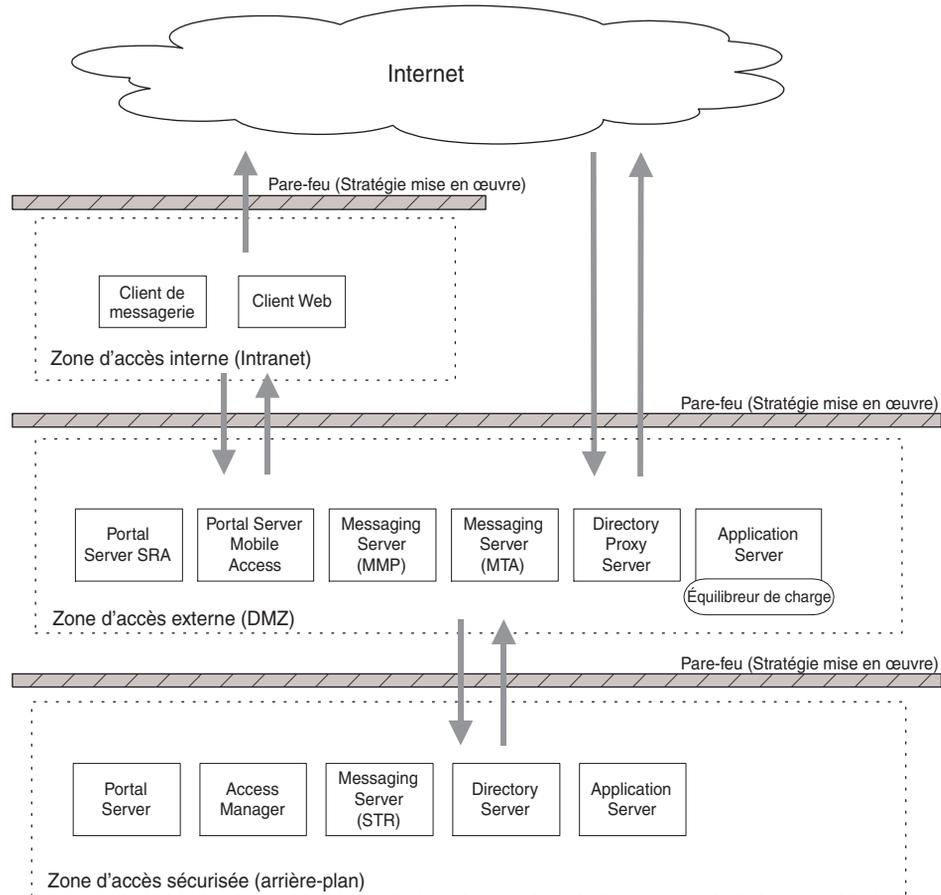
Figure 4-11 Architecture logique de scénario de communications illustrant le deuxième cas d'utilisation



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. La figure ci-dessous illustre les zones d'accès pour le déploiement des composants de Java Enterprise System. Chaque zone d'accès décrit la façon dont les composants fournissent un accès sécurisé à et depuis Internet et l'intranet.

Figure 4-12 Composants logiques placés dans des zones d'accès



Le tableau ci-dessous décrit les zones d'accès présentées dans la [Figure 4-12](#).

Tableau 4-6 Zones d'accès sécurisé et les composants qu'elles contiennent

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 l'accès sécurisé à et depuis Internet en jouant un rôle de tampon de sécurité vers les services d'arrière-plan stratégiques.
Zone d'accès sécurisé (arrière-plan)	Fournit un accès restreint aux services d'arrière-plan stratégiques, accessibles uniquement depuis la zone d'accès externe.

La [Figure 4-12](#) ne représente pas les niveaux logiques décrits dans les exemples précédents, mais détaille 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. Elle fournit une représentation visuelle des règles d'accès à appliquer. Utilisez la conception d'architecture à plusieurs niveaux en association avec la conception 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 de déploiement du cycle de vie de la solution. Vous devez l'associer aux exigences de qualité de service identifiées au cours de la phase d'exigences techniques. L'association de l'architecture logique et des exigences de qualité de service constitue un scénario de déploiement. Celui-ci sert de base à la conception de l'architecture de déploiement, comme l'explique le [chapitre 5](#), « [Conception du déploiement](#) ».

Conception du déploiement

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

Ce chapitre se compose des sections suivantes :

- « À propos de la conception du déploiement », page 76
- « Méthodologie de conception du déploiement », page 79
- « Estimation de la puissance de traitement », page 80
- « Estimation de la puissance de traitement requise pour les transactions sécurisées », page 87
- « Identification des stratégies de disponibilité », page 91
- « Identification des stratégies d'évolutivité », page 100
- « Conception pour une utilisation optimale des ressources », page 105
- « Exemple d'architecture de déploiement », page 107

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 d'exigences techniques du cycle de vie de la solution. Le scénario de déploiement décrit l'architecture logique et les exigences en termes de qualité de service de la 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 entre les exigences de qualité de service, la compensation des compromis et les problèmes liés aux coûts de propriété pour 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 utilisées comme base de travail pour le déploiement. Elles décrivent le matériel à acheter et la structure du réseau. Elles décrivent également les services d'annuaire, y compris l'arbre d'informations sur l'annuaire (DIT, Directory Information Tree) 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 applicables aux 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** : dérivé 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 des données des répertoires et des bases de données existants, des spécifications de structure des répertoires prenant en compte la conception de réplication définie dans l'architecture de déploiement, ainsi que des procédures permettant d'alimenter les répertoires avec le 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 fonctionnels déterminant si une fonctionnalité se comporte selon les prévisions.
 - **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 déploiement. 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 leur nombre jusqu'à ce que le déploiement soit terminé. L'implémentation par phases peut également prévoir le déploiement progressif de logiciels spécifiques.

- **Plan de reprise après incident** : décrit les procédures de restauration du système après une panne. Ces procédures s'appliquent à toutes les 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 affectant la conception du déploiement

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

- **Architecture logique** : celle-ci décrit en détail les services fonctionnels d'une solution proposée et les relations existant entre les composants fournissant ces services. Utilisez-la pour déterminer la meilleure manière de distribuer les services. Un scénario de déploiement présente l'architecture logique associée aux exigences 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) associée aux exigences de qualité de service.
- **Analyse d'utilisation** : l'analyse d'utilisation, développée au cours de la phase d'exigences techniques du cycle de vie de la solution, fournit des informations sur les modèles 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** : ceux-ci sont développés au cours de la phase d'exigences techniques 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 d'une conception de déploiement afin de vous assurer qu'ils sont bien pris en compte.

- **Contrats de niveau de service** : un contrat de niveau de service (SLA, Service Level Agreement) 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 contrat de niveau de service.
- **Coût de propriété total** : lors de la conception du déploiement, vous analysez les solutions potentielles permettant de satisfaire les 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 propriété. Veillez à prendre en compte les compromis exigés par vos décisions 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 du cycle de vie 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

Comme les autres aspects du déploiement, la conception relève plus de l'art que de la science, c'est pourquoi elle ne peut faire l'objet de procédures et de processus détaillés. Les facteurs contribuant à un déploiement réussi sont l'expérience en matière de conception, la connaissance de l'architecture des systèmes, la compréhension du domaine et l'effort de réflexion créative.

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 compenser les compromis de vos décisions de conception afin d'optimiser la solution. La méthodologie utilisée implique généralement les tâches suivantes :

- **Estimation de la puissance de traitement** : la conception du déploiement commence normalement par l'estimation du nombre de processeurs requis pour chaque composant de l'architecture logique. Commencez par les cas d'utilisation correspondant à la charge la plus intense 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 processeurs requis 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 compromis exigés par vos décisions de conception. Par exemple, quel impact les stratégies de disponibilité et d'évolutivité peuvent-elles avoir sur la maintenance du système ? Quels sont les autres coûts de 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 votre conception 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 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 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 processeurs 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 de qualité de service ;

- l'expérience acquise dans le domaine de la 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 une méthode pour prendre en compte les facteurs affectant le résultat final.

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

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

Examinez les interactions entre composants 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 modifiez 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 processeurs estimé en fonction des exigences de sécurité, de disponibilité et d'évolutivité.

Les estimations établies vous permettront de déterminer la puissance de traitement réelle dont vous avez besoin. Elle 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. Celui-ci 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](#) », page 68.

Les estimations de processeurs et de mémoire mentionnées sont arbitraires et fournies à titre d'exemple uniquement. Elles sont fondées sur les données arbitraires sur lesquelles repose 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 :

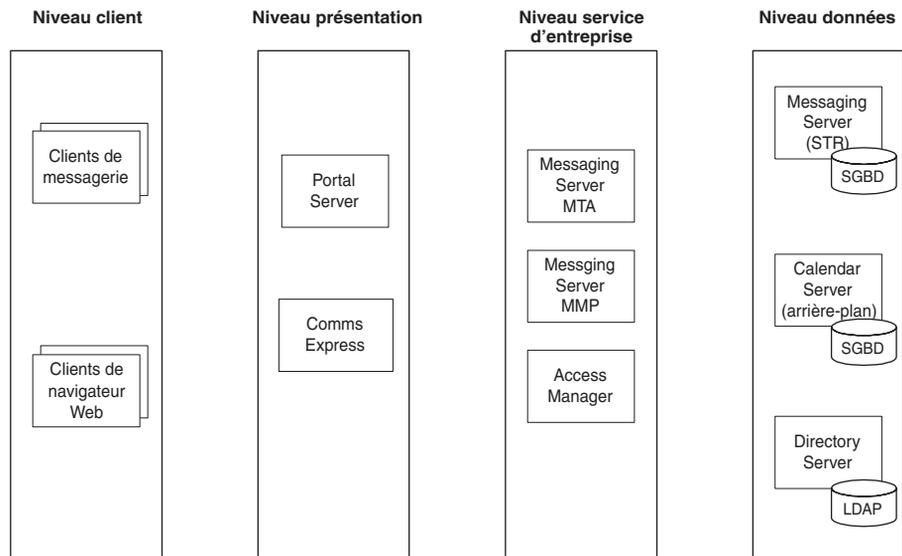
- cas d'utilisation détaillés et analyse d'utilisation fondée sur une analyse d'exploitation exhaustive ;
- exigences de qualité de service déterminée par l'analyse des exigences de l'entreprise ;
- coûts spécifiques et spécifications du matériel de réseau et de traitement ;
- expérience acquise lors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

ATTENTION Les informations présentées dans les exemples ci-après ne fournissent pas de conseils d'implémentation spécifiques. 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 processeurs requis pour les points d'entrée utilisateur

Commencez par évaluer le nombre de processeurs nécessaires pour gérer la charge incombant à chaque composant représentant un point d'entrée utilisateur. La figure ci-dessous illustre l'architecture logique d'un scénario de communications basées sur les identités décrit au [chapitre 4](#), « [Conception logique](#) », page 68.

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 processeurs requis, 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 processeurs pour les composants contenant des points d'entrée utilisateur

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

Inclusion du nombre de processeurs estimé pour les services dépendants

Les composants fournissant des points d'entrée utilisateur doivent être secondés par d'autres Java Enterprise System composants. Pour poursuivre la définition des exigences de performances, incluez les estimations relatives à la prise en charge du support fourni par ces autres composants. 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 dans la section « [Exemples d'architectures logiques](#) », page 63.

Tableau 5-2 Estimation du nombre de processeurs pour les composants de support

Composant	Nombre de processeurs	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 d'agenda pour Communications Express, composant frontal de Calendar Server.
Directory Server	2	Fournit les services d'annuaire LDAP.
Web Server	0	Fournit le support de conteneur Web pour Portal Server et Access Manager. (Aucun cycle de processeur 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 mettiez en évidence 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 cette utilisation de charge de pointe, vous devez modifier les composants fournissant ces services. Le tableau ci-dessous décrit les changements à effectuer.

Tableau 5-3 Modification de l'estimation du nombre de processeurs pour les charges de pointe

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

Modification des estimations pour les autres conditions de charge

Poursuivez votre estimation du nombre de processeurs pour qu'elle tienne compte des autres exigences de qualité de service pouvant avoir un impact sur la charge :

- **Sécurité** : à partir de la phase d'exigences techniques, 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](#) », page 87, explique comment procéder à ces modifications.

- **Réplication des services** : modifiez vos estimations pour qu'elles tiennent compte de la réplication des services à des fins de disponibilité, d'équilibrage de charge et d'évolutivité. La section « [Identification des stratégies de disponibilité](#) », [page 91](#), décrit le concept de dimensionnement dans le cadre des solutions de disponibilité. La section « [Identification des stratégies d'évolutivité](#) », [page 100](#), décrit les solutions permettant un accès efficace aux services d'annuaire.
- **Capacité et évolutivité latentes** : modifiez le nombre de processeurs estimé en fonction de la capacité latente requise pour faire face aux charges importantes 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 processeurs estimé

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

Comptez normalement 2 gigaoctets de mémoire pour chaque processeur. 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 supplémentaire éventuellement ajoutée à des fins de sécurité et de disponibilité. Ces valeurs seront ajoutées dans les sections qui suivent.

Tableau 5-4 Modification de l'estimation du nombre de processeurs pour les composants de support

Composant	Nombre de processeurs	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

Tableau 5-4 Modification de l'estimation du nombre de processeurs pour les composants de support (*Suite*)

Composant	Nombre de processeurs	Mémoire
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 processeurs/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 que Secure Sockets Layer (SSL) ou Transport Layer Security (TLS). Ces transactions exigent généralement une puissance de traitement supplémentaire pour établir une session sécurisée (liaison), puis pour chiffrer et déchiffrer les données transportées. Selon l'algorithme de chiffrement utilisé (à 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 processeurs pour qu'elle tienne 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 de commerce électronique, il n'est pas nécessaire de sécuriser les transactions tant que le client n'a pas terminé ses achats et qu'il n'est pas prêt à procéder au 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 processeurs pour les transactions sécurisées

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

Pour estimer le nombre de processeurs requis pour les transactions sécurisées, effectuez les calculs suivants :

1. Partez d'une estimation du nombre de processeurs requis (voir la section précédente, « [Exemple d'estimation de la puissance de traitement](#) », page 82).
2. Calculez le pourcentage de transactions exigeant un transport sécurisé, puis le nombre de processeurs nécessaires pour ces transactions.
3. Comptez un nombre de processeurs inférieur pour les transactions non sécurisées.
4. Additionnez les valeurs obtenues pour parvenir à une estimation du nombre total de processeurs.
5. Arrondissez cette estimation à une valeur paire.

Le [Tableau 5-5](#) 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 processeurs requis 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 processeurs estimé pour les transactions sécurisées

Étape	Description	Calcul	Résultat
1	Partez d'une estimation de base pour toutes les transactions Portal Server.	Cette estimation, extraite du Tableau 5-3, page 85 , est de 4 processeurs.	-----
2	Calculez l'estimation du nombre de processeurs supplémentaires pour les transactions sécurisées. Supposez que celles-ci exigent une puissance de traitement cinq fois supérieure aux transactions non sécurisées.	10 % des transactions (estimation de base) doivent faire l'objet d'un transport sécurisé : $0,10 \times 4 \text{ processeurs} = 0,4 \text{ processeur}$ Augmentez la puissance de traitement pour les transactions sécurisées selon un facteur de 4 : $4 \times 0,4 = 1,6 \text{ processeur}$	1,6 processeur
3	Comptez un nombre de processeurs inférieur pour les transactions non sécurisées.	90 % des transactions (estimation de base) ne sont pas sécurisées : $0,9 \times 4 \text{ processeurs} = 3,6 \text{ processeurs}$	3,6 processeurs
4	Calculez le nombre total de processeurs estimé pour les transactions sécurisées et non sécurisées.	Transactions sécurisées + non sécurisées = total : $1,6 \text{ processeur} + 3,6 \text{ processeurs} = 5,2 \text{ processeurs}$	5,2 processeurs
5	Arrondissez le résultat à une valeur paire.	$5,2 \text{ processeurs} \Rightarrow 6 \text{ processeurs}$	6 processeurs

Sur la base des calculs effectués pour les transactions sécurisées dans cet exemple, vous devez modifier le nombre total de processeurs estimé dans le [Tableau 5-5, page 89](#) en ajoutant deux processeurs et quatre gigaoctets de mémoire pour parvenir au total ci-dessous pour Portal Server.

Tableau 5-6 Modification du nombre de processeurs estimé pour les transactions Portal Server sécurisées

Composant	Nombre de processeurs	Mémoire
Portal Server	6	12 Go

Matériel dédié au traitement des transactions SSL

Du matériel spécialisé, tel que 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 « liaison » 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 type de 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 liaison, par exemple) et tenir compte du surcroît de complexité qu'il engendre en termes de conception. Cette complexité est due aux 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 meilleure solution répondant 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 (par exemple, 50 %) ?
- 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, décrites à la section « [Identification des stratégies d'évolutivité](#) », page 100. Évitez les solutions complexes exigeant des efforts d'entretien et d'administration importants.

Stratégies de disponibilité

Les stratégies de disponibilité pour 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 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 que la mémoire des messages pour Messaging Server et les données d'agenda pour Calendar Server.

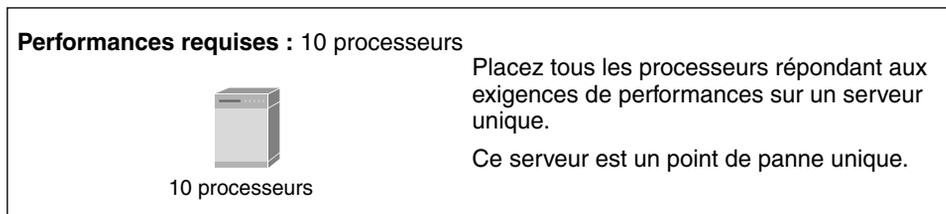
- **Réplication des services** : cette fonction fournit plusieurs sources d'accès aux mêmes données. Directory Server propose de nombreuses stratégies de réplication et de synchronisation pour l'accès aux annuaires LDAP.

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.

Systeme 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 Systeme à serveur unique



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

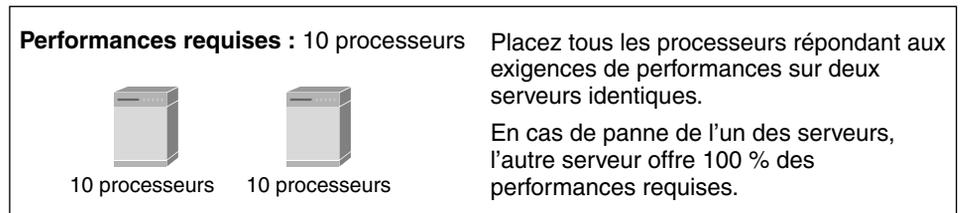
- remplacement et reconfiguration des composants matériels en cours d'exécution du système ;
- possibilité d'exécuter plusieurs applications dans des domaines sécurisés sur le serveur ;
- possibilité de mettre à niveau la capacité, les performances et la configuration des E/S sans redémarrer le système.

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 isolés.

Systèmes à redondance horizontale

Les serveurs redondants parallèles offrant des fonctions d'équilibrage de charge et de basculement prennent en charge plusieurs manières d'améliorer la disponibilité. La figure 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

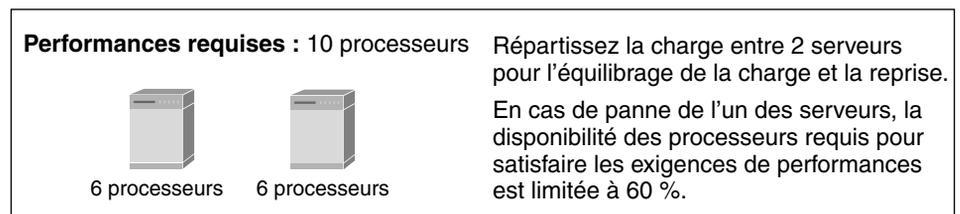


La puissance de traitement de chaque serveur de la [Figure 5-3](#) est identique. Un serveur peut à lui seul satisfaire les exigences de performances. L'autre serveur offre exactement les mêmes performances lorsqu'il est utilisé en tant que serveur de secours.

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.

La figure ci-dessous montre 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

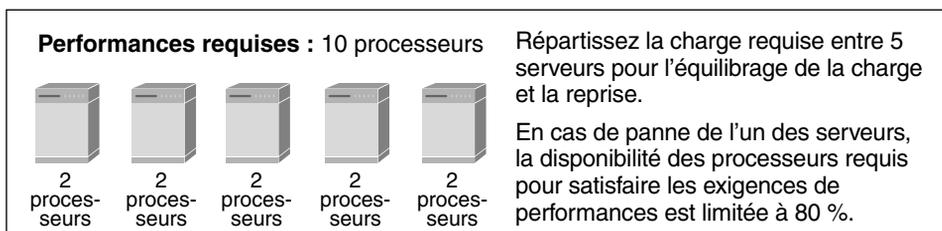


Dans le système représenté à la [Figure 5-4](#), si un serveur tombe en panne, tous les services restent disponibles, mais pas à 100 %. En effet, le second serveur offre 6 processeurs, qui permettent de prendre en charge 60 % de la charge totale (10 processeurs).

Cette conception offre l'avantage d'autoriser une capacité latente (2 processeurs) lorsque les deux serveurs sont disponibles.

La figure 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 [Figure 5-5](#) comportant cinq serveurs, si l'un d'eux tombe en panne, les autres fournissent 8 processeurs, soit 80 % des exigences de performances (10 processeurs). Si vous ajoutez un serveur doté de 2 processeurs, 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 ajoutée 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 (plusieurs serveurs d'entrée de gamme coûtent moins cher qu'un serveur haut de gamme).

Toutefois, plus le nombre de serveurs augmente, plus les coûts d'administration et de maintenance sont élevés. Vous devez également prendre en compte les frais d'hébergement des serveurs dans un centre de données. L'ajout de serveurs supplémentaires 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. 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](#) », page 68. 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 processeurs requis 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 processeurs estimé](#) », page 86.

Tableau 5-7 Modification de l'estimation du nombre de processeurs pour les composants de support

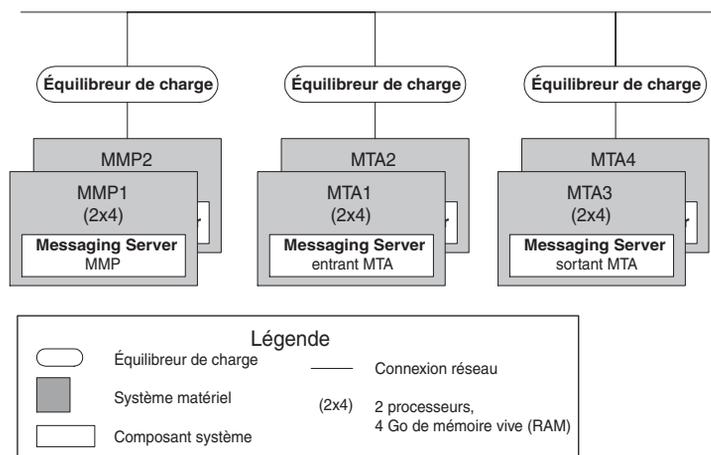
Composant	Nombre de processeurs	Mémoire
Messaging Server (MTA, entrant)	2	4 Go
Messaging Server (MTA, sortant)	2	4 Go
Messaging Server (MMP)	2	4 Go
Messaging Server (Message Store)	2	4 Go

Pour cet exemple, considérez qu’au cours de la phase d’exigences techniques, les exigences de qualité de service suivantes ont été formulées :

- **Disponibilité** : la disponibilité globale du système doit être de 99,99 % (hors périodes d’indisponibilité planifiées). Une panne d’un système individuel 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. La figure ci-dessous illustre le diagramme de réseau correspondant à cette stratégie de disponibilité.

Figure 5-6 Stratégie de disponibilité pour l’exemple Messaging Server



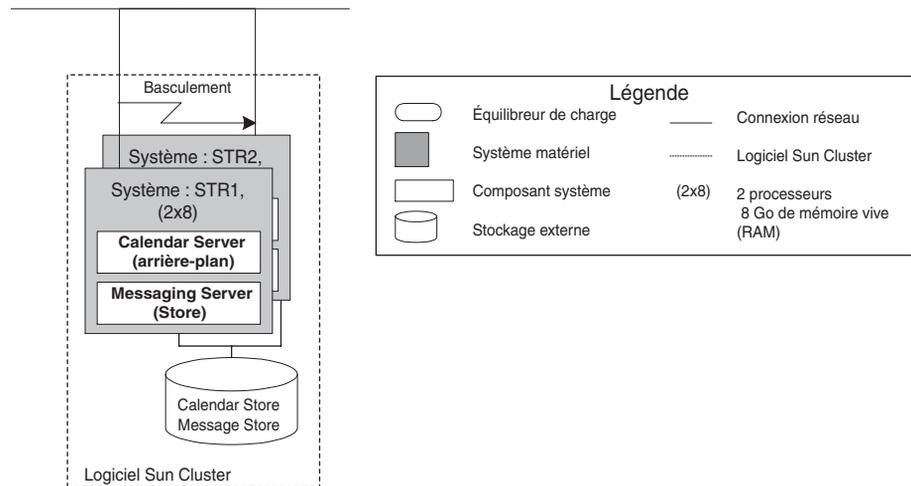
Dans la figure ci-dessous, le nombre de processeurs a doublé par rapport à l’estimation initiale. Cette augmentation est intervenue pour les raisons suivantes :

- 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 ajoute la capacité latente requise.

Exemple de basculement à l'aide du logiciel Sun Cluster

La figure ci-dessous montre un exemple de stratégie de basculement pour le composant d'arrière-plan de Calendar Server et la mémoire de messages de Messaging Server. Le composant d'arrière-plan de 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 processeurs et la mémoire correspondante sont répliqués sur chaque serveur dans Sun Cluster.

Figure 5-7 Conception avec basculement utilisant le logiciel Sun Cluster



Exemple de réplication 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 réplication des services, y compris les suivantes :

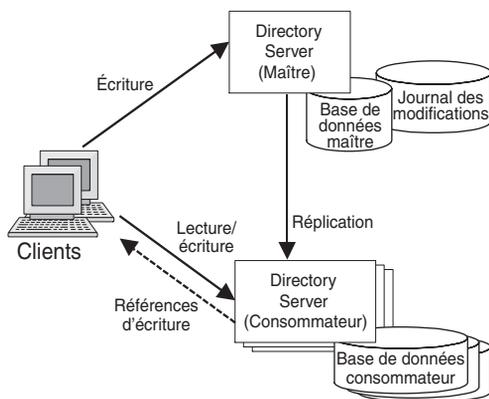
- **Bases de données multiples** : les différentes parties d'une arborescence d'annuaire sont stockées dans des bases de données distinctes.
- **Chaînage et références** : relie les données distribuées dans une même arborescence d'annuaire.
- **Réplication monomaître** : fournit une source centrale pour la base de données maître, qui est ensuite distribuée aux répliques consommateur.
- **Réplication 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 non couvert par ce guide. Les sections « [Réplication monomaître](#) », et « [Réplication multimaître](#) », fournissent une présentation générale des stratégies de réplication de base. Pour plus de détails sur les stratégies de disponibilité pour Directory Server, reportez-vous au manuel *Directory Server Deployment Planning Guide*, <http://docs.sun.com/doc/817-7607>.

Réplication monomaître

La figure ci-dessous représente une stratégie de réplication monomaître illustrant les concepts de réplication de base.

Figure 5-8 Exemple de réplication monomaître



Dans le cadre de la réplication 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. Celui-ci met à jour les bases de données consommateur de manière régulière.

Les avantages de la réplication monomaître sont les suivants :

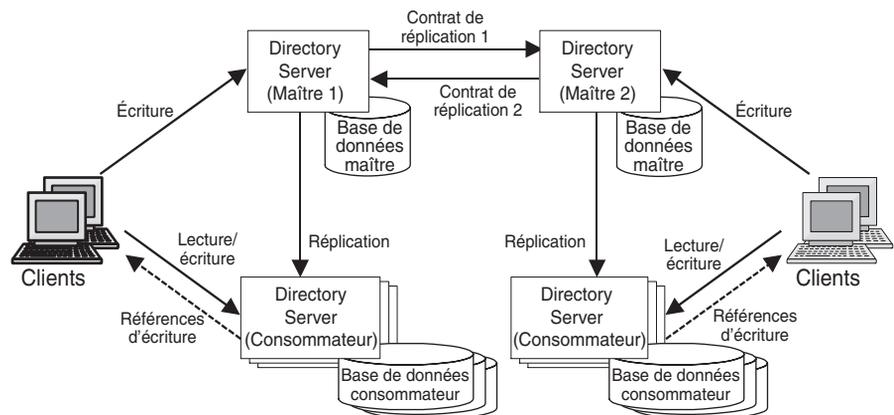
- instance unique de Directory Server optimisée pour les opérations de lecture et d'écriture dans la base de données ;
- nombre quelconque d'instances consommateur de Directory Server optimisées pour les opérations de lecture et de recherche ;
- évolutivité horizontale pour les instances consommateur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

Réplication multimaître

La figure ci-dessous illustre une stratégie de réplication multimaître permettant une distribution globale des accès à l'annuaire.

Avec la réplication multimaître, une ou plusieurs instances de Directory Server gèrent la base de données d'annuaire maître. Chaque maître dispose d'un contrat de réplication 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 réplication 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. Celui-ci met à jour les bases de données consommateur de manière régulière.

Figure 5-9 Exemple de réplication multimaître



La stratégie de réplication multimaître présente les mêmes avantages que la réplication monomaître, auxquels s'ajoute une stratégie de 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 représente la possibilité d'augmenter la capacité du système, généralement via l'ajout de ressources système, sans 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. 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 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. L'idéal serait de 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.

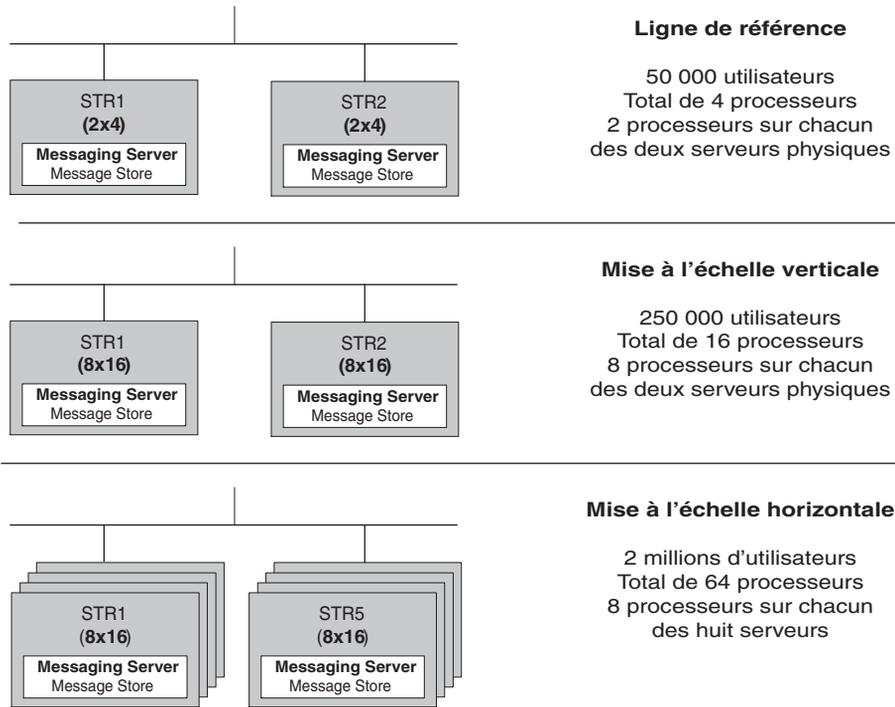
Exemple d'évolutivité

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 horizontale consiste à ajouter des processeurs à un serveur pour gérer les augmentations de charge. La mise à l'échelle verticale consiste à ajouter des serveurs supplémentaires afin de répartir la charge.

Cet exemple se base 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 processeurs, pour un total de quatre processeurs. La figure 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 La [Figure 5-10](#) montre les différences entre les mises à l'échelle verticale et horizontale. Elle ne prend pas en compte les autres facteurs à considérer lors de la mise à l'échelle, tels que l'équilibrage de charge, le basculement et les changements de modèles d'utilisation.

Figure 5-10 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 5-8 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 grappes de disques (RAID).</p> <p>Dédiez 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 largeur de bande et la vitesse d'accès des nœuds du réseau.	<p>Augmentez la largeur de bande.</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 Dans le [Tableau 5-8](#), les catégories de matériel sont répertoriées en fonction de leur vitesse d'accès relative. Les points d'accès lents, tels que les disques, sont donc 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 fréquente 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 dédier les disques à différents types d'opérations d'entrée/sortie. Par exemple, prévoyez des accès séparés pour les opérations de consignation 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 « System Sizing » du manuel *Directory Server Deployment Planning Guide*, <http://docs.sun.com/doc/817-7607>, 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 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 affectant l'utilisation des disques, ainsi que sur la répartition des fichiers entre les disques. Présente également différentes alternatives utilisant des grappes de disques.

Conception pour une utilisation optimale des ressources

La conception du déploiement ne consiste pas simplement à estimer 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 liés à vos décisions de conception pour 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. La mise à l'échelle verticale peut accroître la puissance de traitement à moindre frais mais cette puissance supplémentaire risque de ne pas être utilisée à bon escient par certains services.

Avant de mettre au point votre stratégie de conception, revenez sur vos décisions pour vérifier que l'utilisation des ressources est compensée par les avantages offerts par la solution proposée. Au cours de cette analyse, vous étudierez l'impact des qualités du système dans un domaine sur les qualités du système dans d'autres domaines. Le tableau ci-dessous répertorie certaines qualités du système et les considérations correspondantes en termes de gestion des ressources.

Tableau 5-9 Considérations relatives à la gestion des ressources

Qualité du système	Description
Performances	Si vous adoptez une solution dans laquelle les processeurs sont concentrés sur des serveurs individuels, les services pourront-ils utiliser efficacement la puissance de traitement ? (Par exemple, pour certains services, le nombre de processeurs pouvant être utilisés de manière efficace est limité.)
Capacité latente	<p>Votre stratégie peut-elle gérer les charges dépassant les estimations de performances ?</p> <p>Les charges excessives sont-elles gérées par mise à l'échelle verticale sur les serveurs, équilibrage de charge vers d'autres serveurs, ou les deux ?</p> <p>La capacité latente est-elle suffisante pour gérer les charges de pointe jusqu'à la prochaine étape de mise à l'échelle du déploiement ?</p>
Sécurité	Avez-vous pris en compte les performances supplémentaires nécessaires au traitement des transactions sécurisées ?
Disponibilité	<p>Dans le cas des solutions à redondance horizontale, avez-vous estimé correctement les coûts de maintenance à long terme ?</p> <p>Avez-vous pris en compte les temps d'indisponibilité planifiés nécessaires aux opérations de maintenance du système ?</p> <p>Avez-vous comparé les coûts de serveurs haut de gamme et de serveurs d'entrée de gamme ?</p>
Évolutivité	<p>Avez-vous prévu des étapes de mise à l'échelle du déploiement ?</p> <p>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 ?</p>
Entretien	<p>Votre conception de disponibilité tient-elle compte des coûts d'administration, de contrôle et de maintenance ?</p> <p>Avez-vous envisagé des solutions de délégation permettant aux utilisateurs finals d'effectuer certaines tâches d'administration afin de réduire les coûts ?</p>

Gestion des risques

La plupart des données sur lesquelles se fonde la conception du déploiement, telles que les exigences de qualité de service et l'analyse d'utilisation, ne sont pas empiriques. Il s'agit d'estimations et de projections dérivées 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 tout simplement des erreurs humaines. Avant de conclure 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.

D'autre part, si la puissance de votre système est supérieure aux exigences, cela signifie que 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.

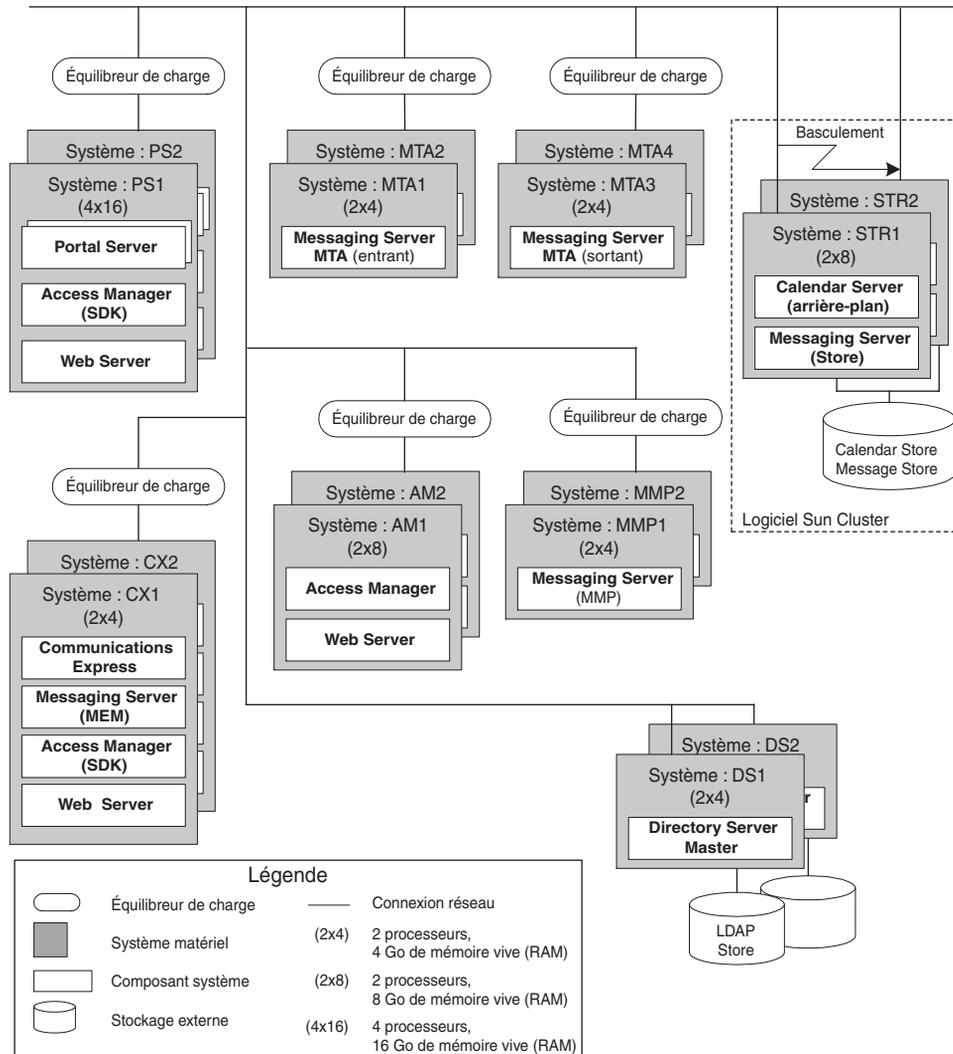
Une utilisation inadaptée des ressources entraîne un échec de la conception car les ressources sous-utilisées auraient pu être employées pour d'autres activités. En outre, les solutions jugées exagérées par les parties prenantes risquent d'être considérées comme ne respectant pas les termes du contrat.

Exemple d'architecture de déploiement

La figure ci-dessous illustre une architecture de déploiement complète correspondant à l'exemple décrit plus tôt dans ce document. Elle fournit un aperçu de la présentation d'une architecture de déploiement.

ATTENTION L'architecture de déploiement illustrée dans la figure ci-dessous est fournie à titre d'exemple uniquement. Elle ne représente pas un déploiement ayant été réellement conçu, mis en place et testé. Elle ne fournit pas de conseils de planification de déploiement.

Figure 5-11 Exemple d'architecture de déploiement



Implémentation d'une conception de déploiement

Lors de la phase d'implémentation du cycle de vie de la solution, 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 déployer la 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 », page 110
- « Installation et configuration du logiciel », page 111
- « Développement de pilotes et de prototypes », page 111
- « Développement de pilotes et de prototypes », page 111
- « Test des pilotes et des prototypes », page 112
- « Passage à un déploiement de production », page 113

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, vous passez à la phase d'implémentation du cycle de vie de la solution. L'implémentation est le résultat d'un ensemble complexe de processus et de procédures dont le succès passe par une préparation minutieuse. Les tâches d'implémentation sont les suivantes

- création de l'infrastructure du réseau et du matériel ;
- installation et configuration des logiciels conformément au plan d'installation ;
- migration des données depuis les 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 d'implémentation.

Installation et configuration du logiciel

L'installation et la configuration de Sun Java™ Enterprise System pour une application d'entreprise distribuée passent par la planification et la coordination de nombreuses tâches et procédures. Au cours de la phase de conception du déploiement, vous créez 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 faire l'objet d'une installation ;
- collecte des informations de configuration relatives à chaque composant de Java Enterprise System à installer.

Le Guide d'installation de Sun Java Enterprise System 2005T1, <http://docs.sun.com/doc/819-0807>, explique comment rassembler les informations nécessaires pour définir un plan d'installation. Ce guide fournit des informations de configuration détaillées et des feuilles de travail documentant ces informations. Il décrit également les scénarios d'installation courants impliquant plusieurs composants de Java Enterprise System. Pour plus d'informations, reportez-vous à la section « Préparation d'une installation » du *Guide d'installation de Sun Java Enterprise System 2005T1*, <http://docs.sun.com/doc/819-0807>.

Développement de pilotes et de prototypes

Les déploiements de Java Enterprise System se classent dans deux catégories : 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 type de déploiement est appelé « 80:20 » (80 % des services sont assurés par Java Enterprise System) et le second, « 20:80 ».

Lors de la phase d'implémentation des déploiements 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 des phases 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 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 une solution proposée résout un problème dans un environnement de test. Dès que 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

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

Idéalement, 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 le débit 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 des 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 pilote doivent généralement 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 dans le cas des déploiements de grande ampleur qui touchent de nombreux utilisateurs.

Vous pouvez d'abord déployer la solution auprès de peu d'utilisateurs, puis augmenter leur nombre jusqu'à ce que le déploiement soit complet. Vous pouvez également déployer un petit nombre de services dans un premier temps, puis continuer avec 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.

Passage à un déploiement de production

A

- Access Manager 57, 84
- Analyse d'utilisation 38
 - Impact sur la conception du déploiement 78
- Application Server 57
- Approbation du projet 76
- Architecture à trois dimensions 54
- Architecture de déploiement 76
 - Exemple 107
- Architecture logique
 - Conception 55
 - Exemple de communications basées sur les identités 69
 - Exemples 63
 - Impact sur la conception du déploiement 78
- Architectures logiques 53

B

- Basculement 91
 - Exemple 97
- Logiciel Sun Cluster 95

C

- Calendar Server 57, 84
- Capacité latente 49
 - Évolutivité 100
- Cas d'utilisation 40
 - Estimation de la puissance de traitement 85
 - Exemple de communications basées sur les identités 69
 - Exemple de Messaging Server 65
 - Impact sur la conception du déploiement 78
- Communications Express 57
- Conception d'architecture à plusieurs niveaux 61
- Conception du déploiement
 - À propos de 76
 - Analyse d'utilisation 78
 - Approbation du projet 76
 - Cas d'utilisation 78
 - Contrats de niveau de service 79
 - Coût de propriété 79
 - Exigences de qualité de service 78
 - Facteurs 78
 - Méthodologie 79
 - Objectifs d'exploitation 79
 - Puissance de traitement 79
 - Réplication des services 80
 - Résultats 76
- Conception logique
 - À propos de 53

- Contraintes d'exploitation 33
 - Coût de propriété 35
 - Établissement du planning 34
 - Limitations budgétaires 34
 - Problèmes de migration 34
- Contrats de niveau de service 33
 - Exigences 51
 - Impact sur la conception du déploiement 79
- Coût de propriété 35
 - Impact sur la conception du déploiement 79
- Culture d'entreprise 31
- Cycle de vie de la solution 22
 - Phase d'analyse d'exploitation 23, 27
 - Phase d'exigences techniques 23, 37
 - Phase d'implémentation 25, 110
 - Phase de conception du déploiement 25, 76
 - Phase de conception logique 24, 53
 - Phase de fonctionnement 26

D

- Dépendances entre composants 57
 - Support de conteneur Web 60
- Déploiements 20:80 20
 - Phase d'implémentation 112
- Déploiements 80:20 20, 111
- Directory Proxy Server 57, 61
- Directory Server 57, 64, 84
 - Réplication monomaître 97, 98
 - Réplication multimaître 97, 99
- Disponibilité
 - Basculement 91
 - Classement par ordre de priorité 44
 - Équilibrage de charge 91
 - Exemples 95
 - Exigence de qualité de service 43
 - Optimisation des ressources 106
 - Réplication des services 92
 - Système à basculement N+1 93
 - Systèmes à redondance horizontale 93
- DMZ
 - Zone d'accès externe 73

- Documentation 12
 - Guide d'installation 58, 111
 - Présentation 12
 - Présentation technique 19, 54, 56

E

- Entretien
 - Exigence de qualité de service 49
 - Optimisation des ressources 106
- Équilibrage de charge 91
 - Exemple 93
- Estimation de la puissance de traitement 79, 80
 - Cas d'utilisation 85
 - Exemple 82
 - Transactions sécurisées 87
- Établissement du planning 34
- Évolutivité
 - Estimation du développement 46
 - Exemple 101
 - Exigence de qualité de service 45
 - Optimisation des ressources 106
 - Stratégies 100
- Exemple de communications basées sur les identités 68
 - Cas d'utilisation 69
 - Estimation de la puissance de traitement 82
- Exemples
 - Architecture de déploiement 107
 - Architecture logique 63
 - Basculement 97
 - Communications basées sur les identités 68
 - Conception de disponibilité 95
 - Directory Server 97
 - Équilibrage de charge 93, 95
 - Estimation de la puissance de traitement 82
 - Estimation de la puissance de traitement requise pour les transactions sécurisées 88
 - Évolutivité 101
 - Exemple d'architecture logique Messaging Server 63
 - Réplication des services 97
 - Réplication monomaître 98
 - Réplication multimaître 99
 - Zones d'accès 72

- Exigences de l'entreprise
 - Contrats de niveau de service 33
 - Culture d'entreprise 31
 - Définition 28
 - Exigences opérationnelles 30
 - Exigences réglementaires 31
 - Identification des besoins des utilisateurs 29
 - Modèles d'utilisation 30
 - Objectifs d'exploitation 28
 - Objectifs de sécurité 32
- Exigences de niveau de service 51
- Exigences de qualité de service 41, 49
 - Impact sur la conception du déploiement 78
 - Rôle dans la conception du déploiement 76
- Exigences opérationnelles 30
- Exigences réglementaires 31
- Exigences techniques
 - Capacité latente 49
 - Disponibilité 43
 - Entretien 49
 - Évolutivité 45
 - Exigences de niveau de service 51
 - Performances 42
 - Sécurité 47

G

- Gestion des risques 107
 - Conception du déploiement 80

I

- Identification des goulots d'étranglement
 - Conception du déploiement 80
- Installation de Java Enterprise System 111
- Instant Messaging 57

J

- Java Enterprise System
 - À propos de 17
 - Architecture à trois dimensions 54
 - Composants 56
 - Composants d'accès 61
 - Dépendances entre composants 57
 - Déploiements 20:80 20
 - Déploiements 80:20 20
 - Installation 111
 - Passage à un déploiement de production 113
 - Problèmes de migration 21
 - Services 20
 - Services personnalisés 20
 - Services système 18

L

- Limitations budgétaires 34
- Logiciel Sun Cluster 95
 - Exemple de basculement 97

M

- Manuel d'exploitation 78
- Message Queue 58
- Messaging Server 58
 - Cas d'utilisation 65
 - Exemple d'architecture logique 63
 - Exemple d'équilibrage de charge 95
 - Message Multiplexor (MMP) 60, 61, 64, 84
 - Message Store (STR) 60, 64, 84
 - Message Transfer Agent (MTA) 60, 64
 - Messenger Express Multiplexor (MEM) 60
 - Services logiques distincts 60
- Modèles d'utilisation 30

N

- Niveau client
 - Modèle d'architecture à plusieurs niveaux 62
- Niveau données
 - Modèle d'architecture à plusieurs niveaux 62
- Niveau présentation
 - Modèle d'architecture à plusieurs niveaux 62
- Niveau services d'entreprise
 - Modèle d'architecture à plusieurs niveaux 62
- Niveaux logiques
 - Modèle d'architecture à plusieurs niveaux 62

O

- Objectifs d'exploitation
 - Définition 28
 - Impact sur la conception du déploiement 79
- Optimisation
 - Accès aux disques 104
 - Utilisation des ressources 105
- Optimisation des ressources
 - Conception du déploiement 80

P

- Performances
 - Exigence de qualité de service 42
 - Identification des goulots d'étranglement 102
 - Optimisation des ressources 106
- Phase d'analyse d'exploitation 23
 - À propos de 27
- Phase d'exigences techniques 23
 - À propos de 37
 - Analyse d'utilisation 38
 - Cas d'utilisation 40
 - Exigences de qualité de service 41
- Phase d'implémentation 25, 111
 - À propos de 110
 - Développement de pilotes et de prototypes 111
- Phase de conception du déploiement 25
- Phase de conception logique 24
- Phase de fonctionnement 26

- Pilotes 111
 - Test 112
- Plan d'installation 77
- Plan de déploiement 77
- Plan de fonctionnement (manuel d'exploitation) 78
- Plan de formation 78
- Plan de gestion des utilisateurs 77
- Plan de migration 77
- Plan de reprise après incident 78
- Plan de test 77
- Planification du déploiement
 - À propos de 21
 - Approche incrémentielle 32
 - Cycle de vie de la solution 22
- Plans d'implémentation 76
- Portal Server 58, 61
 - Mobile Access 61
 - Secure Remote Access 58, 61
- Problèmes de migration 21
 - Contrainte d'exploitation 34
- Prototypes 111
 - Test 112
- Puissance de traitement
 - Estimation 80

Q

- Qualité de service, exigences 41

R

- Réplication des services 80
 - Exemple de Directory Server 97
 - Stratégie de disponibilité 92
- Réplication monomaître 97
 - Exemple 98
- Réplication multimaître 97
 - Exemple 99
- Risques, gestion 107

S

- Scénario de déploiement [53, 73, 76](#)
- Sécurité
 - Estimation de la puissance de traitement [80](#)
 - Exigence de qualité de service [47](#)
 - Optimisation des ressources [106](#)
- SLA [33](#)
- Solaris
 - Patches [14](#)
 - Support technique [14](#)
- Spécifications d'implémentation [76](#)
- Stratégies de disponibilité
 - Identification [91](#)
- Support technique
 - Solaris [14](#)
- Système à basculement N+1 [93](#)
- Systèmes à redondance horizontale [93](#)
- Systèmes à tolérance de pannes [44](#)

T

- Test
 - Pilotes et prototypes [112](#)
 - Tests de charge [112](#)
 - Tests fonctionnels [112](#)
- Tests de charge [112](#)
- Tests fonctionnels [112](#)
- 3 dimensions, architecture [54](#)

W

- Web Server [58, 84](#)

Z

- Zone d'accès externe (DMZ) [73](#)
- Zone d'accès interne (intranet) [73](#)
- Zone d'accès sécurisé [73](#)
- Zones d'accès [72](#)

