



# Sun Java Enterprise System 2005Q4 Handbuch zur Bereitstellungsplanung

---

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

Teilnr.: 819-3454  
Oktober 2005

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Alle Rechte vorbehalten.

Dieses Produkt bzw. Dokument ist urheberrechtlich geschützt und unterliegt Lizenzen, die die Verwendung, Vervielfältigung, Weitergabe und Dekompilierung einschränken. Die vollständige oder teilweise Reproduktion dieses Produkts bzw. Dokuments ist in keiner Form und auf keine Weise ohne vorherige schriftliche Genehmigung durch Sun oder gegebenenfalls durch seine Lizenzgeber gestattet. Software von Drittanbietern, einschließlich der Schriftart-Technologie, ist urheberrechtlich geschützt und von Sun-Vertragspartnern lizenziert.

Teile dieses Produkts können von Berkeley BSD Systems abgeleitet sein, lizenziert durch die University of California. UNIX ist ein eingetragenes Warenzeichen in den USA und anderen Ländern und exklusiv durch X/Open Company, Ltd. lizenziert.

Sun, Sun Microsystems, das Sun-Logo, docs.sun.com, AnswerBook, AnswerBook2, Java, und Solaris sind Warenzeichen oder eingetragene Warenzeichen von Sun Microsystems, Inc., in den USA und anderen Ländern. Alle SPARC-Warenzeichen werden unter Lizenz verwendet und sind Warenzeichen oder eingetragene Warenzeichen von SPARC International, Inc., in den USA und anderen Ländern. Produkte, die das SPARC-Warenzeichen aufweisen, basieren auf der von Sun Microsystems, Inc., entwickelten Architektur.

Die grafische Benutzeroberfläche von OPEN LOOK und Sun™ wurde von Sun Microsystems, Inc., für die entsprechenden Benutzer und Lizenznehmer entwickelt. Sun erkennt die Pionierleistung von Xerox bei der Ausarbeitung und Entwicklung des Konzepts von visuellen oder grafischen Benutzeroberflächen für die Computerindustrie an. Sun ist Inhaber einer einfachen Lizenz von Xerox für die Xerox Graphical User Interface (grafische Benutzeroberfläche von Xerox). Mit dieser Lizenz werden auch die Sun-Lizenznehmer abgedeckt, die grafische OPEN LOOK-Oberflächen implementieren und sich ansonsten an die schriftliche Sun-Lizenzvereinbarung halten.

Rechte der US-Regierung – Kommerzielle Software. Regierungsbenutzer unterliegen der standardmäßigen Lizenzvereinbarung von Sun Microsystems, Inc., sowie den anwendbaren Bestimmungen der FAR und ihrer Zusätze.

DIE DOKUMENTATION WIRD IN DER VERFÜGBAREN FORM ZUR VERFÜGUNG GESTELLT UND ALLE AUSDRÜCKLICHEN ODER STILLSCHWEIGENDEN BEDINGUNGEN, ANGABEN UND GARANTIEN, INKLUSIVE ALLER STILLSCHWEIGENDEN GARANTIEN BEZÜGLICH HANDELSÜBLICHKEIT, EIGNUNG ZU EINEM BESTIMMTEN ZWECK ODER MÄNGELGEWÄHR, SIND VON DER HAFTUNG AUSGESCHLOSSEN, AUSSER EIN SOLCHER AUSSCHLUSS WIRD ALS RECHTSWIDRIG BEFUNDEN.

---

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

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

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

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

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPENDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



051220@13215



# Inhalt

---

|  |           |
|--|-----------|
| <b>Vorwort</b>                                     | <b>11</b> |
| <b>1 Einführung in die Bereitstellungsplanung</b>  | <b>17</b> |
| Informationen zu Java Enterprise System            | 17        |
| Systemdienste                                      | 17        |
| Integrierte Dienste und benutzerdefinierte Dienste | 19        |
| Migration auf Java Enterprise System               | 20        |
| Informationen zur Bereitstellungsplanung           | 21        |
| Lebenszyklus einer Lösung                          | 21        |
| Phase der Geschäftsanalyse                         | 23        |
| Phase der technischen Anforderungen                | 23        |
| Phase des logischen Konzepts                       | 24        |
| Phase des Bereitstellungskonzepts                  | 24        |
| Implementierungsphase                              | 25        |
| Betriebsphase                                      | 26        |
| <b>2 Geschäftsanalyse</b>                          | <b>27</b> |
| Informationen zur Geschäftsanalyse                 | 27        |
| Definieren der Geschäftsanforderungen              | 28        |
| Festlegen der Geschäftsziele                       | 28        |
| Kenntnis der Benutzerbedürfnisse                   | 29        |
| Kenntnis der Unternehmenskultur                    | 30        |
| Schrittweise Annäherung                            | 32        |
| Kenntnis der Vereinbarungen auf Dienstebene        | 32        |
| Definieren der geschäftlichen Einschränkungen      | 33        |
| Migrationsprobleme                                 | 33        |

|          |   |           |
|----------|---|-----------|
|          | Planungsmandate   | 33        |
|          | Budgeteinschränkungen   | 34        |
|          | Gesamtkosten  | 34        |
| <b>3</b> | <b>Technische Anforderungen</b>                                 | <b>35</b> |
|          | Informationen zu technischen Anforderungen                      | 35        |
|          | Anwendungsanalyse   | 36        |
|          | Anwendungsfälle   | 38        |
|          | Dienstqualitätsanforderungen                                    | 39        |
|          | Leistung  | 40        |
|          | Verfügbarkeit   | 41        |
|          | Skalierbarkeit  | 43        |
|          | Sicherheitsanforderungen  | 45        |
|          | Latente Kapazität   | 46        |
|          | Wartungseignungsanforderungen                                   | 47        |
|          | Anforderungen an die Dienstebene                                | 48        |
| <b>4</b> | <b>Logisches Konzept</b>  | <b>49</b> |
|          | Informationen zu logischen Architekturen                        | 49        |
|          | Entwurf einer logischen Architektur                             | 51        |
|          | Java Enterprise System-Komponenten                              | 51        |
|          | Komponentenabhängigkeiten                                       | 52        |
|          | Webcontainer-Unterstützung                                      | 55        |
|          | Von Messaging Server bereitgestellte logisch eindeutige Dienste | 55        |
|          | Zugriffskomponenten   | 56        |
|          | Entwurf einer mehrschichtigen Architektur                       | 56        |
|          | Beispiel für logische Architekturen                             | 58        |
|          | Messaging Server-Beispiel                                       | 58        |
|          | Identitätsbasiertes Kommunikationsbeispiel                      | 63        |
|          | Zugriffszonen   | 67        |
|          | Bereitstellungsszenario   | 68        |
| <b>5</b> | <b>Bereitstellungskonzept</b>                                   | <b>69</b> |
|          | Informationen zu Bereitstellungskonzepten                       | 69        |
|          | Projektgenehmigung  | 70        |
|          | Bereitstellungskonzeptausgaben                                  | 70        |
|          | Faktoren mit Auswirkung auf das Bereitstellungskonzept          | 72        |

|  |            |
|--|------------|
| Methodik des Bereitstellungskonzepts                             | 73         |
| Einschätzen von Prozessoranforderungen                           | 74         |
| Beispiel für das Einschätzen von Prozessoranforderungen          | 75         |
| Einschätzen von Prozessoranforderungen für sichere Transaktionen | 81         |
| CPU-Einschätzungen für sichere Transaktionen                     | 82         |
| Spezielle Hardware zur Handhabung von SSL-Transaktionen          | 83         |
| Ermitteln von Verfügbarkeitsstrategien                           | 84         |
| Verfügbarkeitsstrategien   | 84         |
| Beispiele für Verfügbarkeitskonzepte                             | 88         |
| Festlegen von Strategien hinsichtlich der Skalierbarkeit         | 93         |
| Latente Kapazität  | 93         |
| Skalierbarkeitsbeispiel  | 94         |
| Identifizieren von Leistungsengpässen                            | 95         |
| Optimieren des Festplattenzugriffs                               | 97         |
| Konzeption für optimale Ressourcennutzung                        | 98         |
| Verwalten von Risiken  | 99         |
| Beispiel für eine Bereitstellungsarchitektur                     | 100        |
| <b>6 Implementierung eines Bereitstellungskonzepts</b>           | <b>103</b> |
| Informationen zur Implementierung des Bereitstellungskonzepts    | 103        |
| Installation und Konfiguration der Software                      | 104        |
| Entwickeln von Pilotbereitstellungen und Prototypen              | 105        |
| Testen von Pilot- und Prototypbereitstellungen                   | 106        |
| Einsetzen einer Produktionsbereitstellung                        | 107        |
| <b>Index</b>   | <b>109</b> |



# Tabellen

---

|                    |   |    |
|--------------------|---|----|
| <b>TABELLE 1-1</b> | Java Enterprise System-Dienstkategorien   | 18 |
| <b>TABELLE 3-1</b> | Faktoren der Anwendungsanalyse  | 37 |
| <b>TABELLE 3-2</b> | Systemqualitäten mit Auswirkung auf QoS-Anforderungen                                     | 39 |
| <b>TABELLE 3-3</b> | Unvorhergesehene Ausfallzeit für ein System, das ganzjährig (8.760 Stunden) verfügbar ist | 42 |
| <b>TABELLE 3-4</b> | Verfügbarkeit von Diensten nach Priorität   | 43 |
| <b>TABELLE 3-5</b> | Skalierbarkeitsfaktoren   | 44 |
| <b>TABELLE 3-6</b> | Aspekte für die Wartungseignungsanforderungen   | 47 |
| <b>TABELLE 4-1</b> | Java Enterprise System-Komponentenabhängigkeiten  | 53 |
| <b>TABELLE 4-2</b> | Messaging Server-Konfigurationen  | 55 |
| <b>TABELLE 4-3</b> | Java Enterprise System-Komponenten, die Remote-Zugriff bieten                             | 56 |
| <b>TABELLE 4-4</b> | Logische Schichten in einer mehrschichtigen Architektur                                   | 57 |
| <b>TABELLE 4-5</b> | Komponenten in der logischen Architektur von Messaging Server                             | 59 |
| <b>TABELLE 4-6</b> | Sichere Zugriffszonen und darin platzierte Komponenten                                    | 68 |
| <b>TABELLE 5-1</b> | CPU-Einschätzungen für Komponenten mit zugriffsbezogenen Benutzereinstiegspunkten         | 77 |
| <b>TABELLE 5-2</b> | CPU-Einschätzungen für unterstützende Komponenten   | 78 |
| <b>TABELLE 5-3</b> | Anpassungen der CPU-Einschätzung hinsichtlich der Spitzenauslastung                       | 79 |
| <b>TABELLE 5-4</b> | Anpassungen von CPU-Einschätzungen für unterstützende Komponenten                         | 80 |
| <b>TABELLE 5-5</b> | Ändern von CPU-Einschätzungen für sichere Transaktionen                                   | 82 |
| <b>TABELLE 5-6</b> | Anpassungen von CPU-Einschätzungen für unterstützende Komponenten                         | 89 |
| <b>TABELLE 5-7</b> | Datenzugriffspunkte   | 96 |
| <b>TABELLE 5-8</b> | Überlegungen hinsichtlich Ressourcenverwaltung  | 98 |





# Abbildungen

---

|                       |   |     |
|-----------------------|---|-----|
| <b>ABBILDUNG 1-1</b>  | Lebenszyklus einer Lösung   | 22  |
| <b>ABBILDUNG 4-1</b>  | Java Enterprise System-Komponenten                                      | 52  |
| <b>ABBILDUNG 4-2</b>  | Java Enterprise System-Komponentenabhängigkeiten                        | 54  |
| <b>ABBILDUNG 4-3</b>  | Mehrschichtiges Architekturmodell                                       | 57  |
| <b>ABBILDUNG 4-4</b>  | Logische Architektur für die Messaging Server-Bereitstellung            | 59  |
| <b>ABBILDUNG 4-5</b>  | In Zugriffszonen platzierte logische Komponenten                        | 67  |
| <b>ABBILDUNG 5-1</b>  | Logische Architektur für ein identitätsbasiertes Kommunikationsszenario | 77  |
| <b>ABBILDUNG 5-2</b>  | System mit nur einem Server   | 85  |
| <b>ABBILDUNG 5-3</b>  | N+1-Failover-System mit zwei Servern                                    | 86  |
| <b>ABBILDUNG 5-4</b>  | Lastenausgleich plus Failover zwischen zwei Servern                     | 87  |
| <b>ABBILDUNG 5-5</b>  | Verteilung der Last zwischen $n$ Servern                                | 87  |
| <b>ABBILDUNG 5-6</b>  | Failover-Konzept mit Sun Cluster-Software                               | 90  |
| <b>ABBILDUNG 5-7</b>  | Beispiel für eine Einzelmaster-Replikation                              | 91  |
| <b>ABBILDUNG 5-8</b>  | Beispiel für eine Multimaster-Replikation                               | 92  |
| <b>ABBILDUNG 5-9</b>  | Beispiele für horizontale und vertikale Skalierung                      | 95  |
| <b>ABBILDUNG 5-10</b> | Beispiel für eine Bereitstellungsarchitektur                            | 101 |



# Vorwort

---

Das Handbuch *Sun Java Enterprise System 2005Q4 Handbuch zur Bereitstellungsplanung* bietet eine Einführung in die Planung und in die Konzeptentwicklung von Bereitstellungslösungen für Unternehmen, die auf Sun Java™ Enterprise System basieren. In diesem Handbuch werden die Grundlagen und Prinzipien der Bereitstellungsplanung und des Bereitstellungskonzepts sowie der Lebenszyklus einer Lösung beschrieben, der die unterschiedlichen Phasen und Aufgaben eines Bereitstellungsplanungsprojekts umfasst. Darüber hinaus werden allgemeine Beispiele und Strategien angegeben, die Sie bei der Planung der unternehmensweiten Bereitstellung von Lösungen mit Java Enterprise System (Java ES) anwenden können.

---

## Zielgruppe dieses Handbuchs

Dieses Handbuch richtet sich in erster Linie an Bereitstellungsarchitekten und Unternehmensplaner, die für die Analyse- und Konzepterstellung für Unternehmensbereitstellungen verantwortlich sind. Des Weiteren eignet sich dieses Handbuch auch für Systemintegratoren und andere Benutzer, die für die Konzipierung und Implementierung verschiedener Aspekte einer Unternehmensanwendung verantwortlich sind.

---

## Bevor Sie dieses Handbuch lesen

Im vorliegenden Handbuch wird angenommen, dass Sie mit dem Konzept und der Installation von Unternehmensanwendungen und dem Inhalt des Handbuchs *Sun Java Enterprise System 2005Q4 Technischer Überblick* vertraut sind.

---

## Aufbau des Handbuchs

Dieses Handbuch legt einen Lösungslebenszyklus zugrunde, anhand dessen die verschiedenen Phasen der Bereitstellungsplanung erläutert werden. In [Kapitel 1](#) wird der Lebenszyklus einer Lösung beschrieben.

---

## Java ES-Dokumentation

In der Dokumentation zu Java ES wird die Bereitstellungsplanung und die Systeminstallation beschrieben. Der URL für die Systemdokumentation lautet <http://docs.sun.com/coll/1286.1>. Um eine Einführung in Java ES zu erhalten, lesen Sie die in der folgenden Tabelle aufgelisteten Handbücher. Lesen Sie die Handbücher in der Reihenfolge, in der sie aufgelistet sind.

**TABELLE P-1** Java Enterprise System-Dokumentation

| Dokument   | Inhalt  |
|--|---|
| <i>Sun Java Enterprise System 2005Q4 Versionshinweise</i>                    | Enthält die neuesten Informationen über das Java ES, einschließlich bekannter Probleme. Zu den einzelnen Komponenten sind außerdem eigene Versionshinweise vorhanden.   |
| <i>Übersicht über die Dokumentation zu Sun Java Enterprise System 2005Q4</i> | Enthält Beschreibungen der Dokumentation zu Java ES. Es sind Verknüpfungen zur Dokumentation für die Komponenten enthalten.   |
| <i>Sun Java Enterprise System 2005Q4 Technischer Überblick</i>               | Bietet eine Einführung in die technischen und konzeptionellen Grundlagen von Java ES. Beschreibt die Komponenten, die Architektur sowie Prozesse und Funktionen.  |
| <i>Sun Java Enterprise System 2005Q4 Handbuch zur Bereitstellungsplanung</i> | Bietet eine Einführung in die Planung und die Konzeptentwicklung von Bereitstellungslösungen für Unternehmen, die auf Java ES basieren. Beschreibt die Grundlagen und Prinzipien der Bereitstellungsplanung und des Bereitstellungskonzepts sowie den Lebenszyklus einer Lösung und bietet allgemeine Beispiele und Strategien, die Sie bei der Planung der unternehmensweiten Bereitstellung von Lösungen mit Java ES anwenden können. |

**TABELLE P-1** Java Enterprise System-Dokumentation (Fortsetzung)

| Dokument  | Inhalt  |
|---|---|
| <i>Java ES System 2005Q4 Handbuch zur Installationsplanung</i>                          | Unterstützt Sie bei der Entwicklung der Implementierungsspezifikationen hinsichtlich Hardware, Betriebssystem und Netzwerkaspekte Ihrer Java ES-Bereitstellung. Beschreibt Besonderheiten, wie beispielsweise Komponentenabhängigkeiten, die in Ihrem Installations- und Konfigurationsplan berücksichtigt werden müssen. |
| <i>Sun Java Enterprise System 2005Q4 Installationshandbuch für UNIX</i>                 | Führt sich durch den Installationsvorgang von Java ES für das Solaris- oder Linux-Betriebssystem. Zeigt auf, wie Sie die Komponenten nach der Installation konfigurieren und wie Sie überprüfen, dass die konfigurierten Komponenten ordnungsgemäß funktionieren.   |
| <i>Java ES System 2005Q4 Installation-Verweis</i>                                       | Bietet zusätzliche Informationen zu Konfigurationsparametern, enthält Arbeitsblätter, die Sie für Ihren Konfigurationsplan verwenden können, sowie Referenzmaterialien, wie beispielsweise Standardverzeichnisse und standardmäßige Anschlussnummern.   |
| <i>Sun Java Enterprise System 2005Q1 Deployment Example Series: Evaluation Scenario</i> | Beschreibt die Installation von Java ES auf einem System, das Einrichten von Core-Diensten, gemeinsam genutzten Diensten sowie Netzwerkdiensten und das Einrichten von Benutzerkonten, mit denen der Zugriff auf die eingerichteten Dienste möglich ist.  |
| <i>Sun Java Enterprise System 2005Q4 Aufrüstungshandbuch</i>                            | Bietet Anweisungen für die Aufrüstung von Java ES für die Solaris- oder die Linux-Betriebsumgebung.   |
| <i>Sun Java Enterprise System Glossary</i>  | Definiert die Begriffe, die in der Java ES-Dokumentation verwendet werden.  |

---

## Typografische Konventionen

In der folgenden Tabelle werden die in diesem Handbuch verwendeten typografischen Änderungen beschrieben.

TABELLE P-2 Typografische Konventionen

| Schriftart       | Bedeutung  | Beispiel   |
|------------------|--|--|
| AaBbCc123        | Befehlsnamen, Dateinamen, Verzeichnisnamen und Bildschirmcomputerausgaben  | Bearbeiten Sie die Datei <code>.login</code> .<br>Verwenden Sie <code>ls -a</code> zum Auflisten aller Dateien.<br><code>machine_name% you have mail.</code> |
| <b>AaBbCc123</b> | Eingabe durch den Benutzer (im Gegensatz zur Computerausgabe auf dem Bildschirm).  | <code>Computername% su</code><br>Password:   |
| <i>AaBbCc123</i> | Ein Platzhalter in einem Befehl oder Pfadnamen, der mit einem wirklichen Namen oder einem Wert ersetzt werden soll.                            | Der Befehl zum Entfernen einer Datei lautet <code>rm <i>Dateiname</i></code> .   |
| <i>AaBbCc123</i> | Buchtitel, neue Begriffe, Wörter, die betont werden sollen. (Beachten Sie, dass einige betonte Elemente online fettgedruckt angezeigt werden.) | Lesen Sie Kapitel 6 im <i>User's Guide</i> .<br>Ein <i>Cache</i> ist eine lokal gespeicherte Kopie.<br>Speichern Sie die Datei <i>nicht</i> .                |

---

## Shell-Eingabeaufforderungen in Befehlsbeispielen

In der folgenden Tabelle werden die standardmäßigen Systemeingabeaufforderungen und Superuser-Eingabeaufforderungen beschrieben.

TABELLE P-3 Shell-Eingabeaufforderungen

| Shell  | Eingabeaufforderung        |
|--|----------------------------|
| C-Shell unter UNIX und Linux                               | <code>machine_name%</code> |
| C-Shell-Superuser unter UNIX und Linux                     | <code>machine_name#</code> |
| Bourne-Shell und Korn-Shell unter UNIX und Linux           | <code>\$</code>            |
| Bourne-Shell und Korn-Shell-Superuser unter UNIX und Linux | <code>#</code>             |
| Microsoft Windows-Befehlszeile                             | <code>C:\</code>           |

---

## Symbolkonventionen

In der folgenden Tabelle werden die in diesem Handbuch verwendeten Symbole beschrieben.

**TABELLE P-4** Symbolkonventionen

| Symbol | Beschreibung  | Beispiel                     | Bedeutung  |
|--------|---|------------------------------|--|
| [ ]    | Enthält optionale Argumente und Befehlsoptionen.                                | ls [-l]                      | Die Option -l ist nicht erforderlich.  |
| {   }  | Enthält verschiedene Möglichkeiten für eine erforderliche Befehlsoption.        | -d {y n}                     | Für die Option -d muss entweder das Argument y oder das Argument n verwendet werden.                         |
| \${ }  | Gibt eine Variablenreferenz an.   | \${com.sun.javaRoot}         | Referenziert den Wert der com.sun.javaRoot-Variablen.  |
| -      | Gibt eine Kombination aus mehreren Tasten an.                                   | STRG-A                       | Halten Sie die Taste STRG gedrückt, während Sie die Taste A drücken.   |
| +      | Gibt die aufeinander folgende Betätigung mehrerer Tasten an.                    | STRG+A+N                     | Drücken Sie die Taste STRG, lassen Sie sie danach los und drücken Sie anschließend die nachfolgenden Tasten. |
| →      | Zeigt die Auswahl eines Menüelements in einer grafischen Benutzeroberfläche an. | „Datei“ → „Neu“ → „Vorlagen“ | Wählen Sie im Menü „Datei“ die Option „Neu“ aus. Wählen Sie im Untermenü „Neu“ die Option „Vorlagen“ aus.    |

---

## Online-Zugriff auf Sun-Ressourcen

Auf der Website docs.sun.com<sup>SM</sup> können Sie online auf die technische Dokumentation von Sun zugreifen. Sie können das Archiv durchsuchen oder nach einem bestimmten Handbuchtitel oder Thema suchen. Die Handbücher sind als Online-Dateien im PDF-Format (Portable Document Format) oder HTML-Format (Hypertext Markup Language) verfügbar. Beide Formate können von unterstützender Technologie für Benutzer mit Behinderungen gelesen werden.

Unter der Adresse <http://www.sun.com> erhalten Sie Zugriff auf folgende Sun-Ressourcen:

- Downloads von Sun-Produkten
- Services und Lösungen
- Support (einschließlich Patches und Updates)
- Schulung
- Informationen über Sun
- Communities (z. B. Sun Developer Network)

---

## Verweise auf Websites von Drittanbietern

In diesem Dokument sind Verweise auf Websites von Drittanbietern enthalten, auf denen weitere mit dem Thema verwandte Informationen bereitgestellt werden.

---

**Hinweis** – Sun ist nicht verantwortlich für die Verfügbarkeit der in diesem Dokument erwähnten Websites von Drittanbietern. Sun unterstützt keine Inhalte, Werbung, Produkte oder sonstige Materialien, die auf oder über solche Websites oder Ressourcen verfügbar sind, und übernimmt keine Verantwortung oder Haftung dafür. Sun ist nicht verantwortlich oder haftbar für tatsächliche oder vermeintliche Schäden oder Verluste, die durch oder in Verbindung mit der Verwendung von über solche Websites oder Ressourcen verfügbaren Inhalten, Waren oder Dienstleistungen bzw. dem Vertrauen darauf entstanden sind oder angeblich entstanden sind.

---

---

## Kommentare sind willkommen

Sun möchte seine Dokumentation laufend verbessern. Ihre Kommentare und Vorschläge sind daher immer willkommen. Zum Mitteilen Ihrer Kommentare rufen Sie <http://docs.sun.com> auf und klicken Sie auf „Kommentare senden“. Geben Sie im Online-Formular den Dokumenttitel und die Teilenummer an. Die Teilenummer ist eine 7-stellige oder 9-stellige Zahl, die Sie auf der Titelseite des Handbuchs oder im URL der Online-Version des Handbuchs finden. Die Teilenummer für dieses Handbuch lautet beispielsweise 819-3454.



# Einführung in die Bereitstellungsplanung

---

Dieses Kapitel bietet eine Kurzübersicht über Sun Java™ Enterprise System (Java ES), enthält Ausführungen zu Bereitstellungsplanungskonzepten und gibt eine Einführung zum Lösungslebenszyklus, in dem die unterschiedlichen Schritte in der Planung und im Entwurf von Softwaresystemen in Unternehmen beschrieben werden. Dieses Kapitel enthält die folgenden Abschnitte:

- „Informationen zu Java Enterprise System“ auf Seite 17
- „Informationen zur Bereitstellungsplanung“ auf Seite 21

---

## Informationen zu Java Enterprise System

Java Enterprise System ist eine Software-Infrastruktur, die eine komplette Gruppe von Middleware-Diensten zur Unterstützung von Unternehmensanwendungen bereitstellt, die über ein Netzwerk oder in einer Internetumgebung verteilt wurden. Die Java Enterprise System-Komponenten für die Bereitstellung der Dienste werden mit einem gängigen Installationsprogramm installiert und mit einer allgemeinen Gruppe freigegebener Bibliotheken synchronisiert. Sie nutzen gemeinsam ein integriertes System für die Benutzeridentitäts- und Sicherheitsverwaltung.

### Systemdienste

Die wesentlichen von Java Enterprise System bereitgestellten Komponenten können in folgende Kategorien unterteilt werden:

- **Portaldienste.** Diese Dienste bieten mobilen Mitarbeitern, Telearbeitern, Schulungsmitarbeitern, Geschäftspartnern, Lieferanten und Kunden die Möglichkeit, von einem beliebigen Standort außerhalb des Unternehmensnetzwerks aus über das Internet sicher auf ihr persönliches

Unternehmensportal zuzugreifen. Diese Dienste ermöglichen Benutzer-Communities jederzeit und von beliebigen Standorten aus den Zugriff auf das Portal und ermöglichen somit Personalisierung, Aggregation, Sicherheit, Integration, mobilen Zugriff und Suchfunktionen.

- **Kommunikations- und Zusammenarbeitsdienste.** Diese Dienste ermöglichen den sicheren Austausch von Informationen innerhalb verschiedener Benutzer-Communities. Es stehen spezifische Funktionen, wie beispielsweise Messaging, Zusammenarbeit in Echtzeit sowie Kalenderplanungsfunktionen, im Zusammenhang mit der Unternehmensumgebung des jeweiligen Benutzers zur Verfügung.
- **Netzwerkidentitäts- und Sicherheitsdienste.** Diese Dienste verbessern die Sicherheit und den Schutz von wichtigen Unternehmensinformationen, indem sie die Durchsetzung entsprechender Zugriffssteuerungsrichtlinien über alle Communities, Anwendungen und Dienste hinweg auf globaler Ebene gewährleisten. Diese Dienste verwenden ein zentrales Repository zum Speichern und Verwalten von Identitätsprofilen, Zugriffsrechten sowie von Informationen zu Anwendungen und Netzwerkressourcen.
- **Web- und Anwendungsdienste.** Diese Dienste ermöglichen bereitgestellten Komponenten die Kommunikation und unterstützen die Entwicklung, Bereitstellung und Verwaltung von Anwendungen für eine große Anzahl unterschiedlicher Server, Clients und Geräte. Die Dienste basieren auf der Enterprise Edition (J2EE™)-Technologie für die Java 2-Plattform.
- **Verfügbarkeitsdienste.** Diese Dienste bieten eine fast kontinuierliche Verfügbarkeit und Skalierbarkeit für Anwendungen und Webdienste.

In der folgenden Tabelle werden die oben aufgeführten Dienstkategorien und die Java Enterprise System-Komponenten aufgelistet, die die Dienste für die jeweiligen Kategorien bereitstellen.

**TABELLE 1–1** Java Enterprise System-Dienstkategorien

| Dienstkategorie | Java Enterprise System-Komponenten  |
|-----------------|---|
| Zugangsdienste  | Portal Server<br>Portal Server Secure Remote Access<br>Access Manager<br>Directory Server<br>Application Server oder Web Server |

**TABELLE 1–1** Java Enterprise System-Dienstkategorien (Fortsetzung)

| Dienstkategorie                               | Java Enterprise System-Komponenten |
|---|------------------------------------|
| Kommunikations- und<br>Zusammenarbeitsdienste | Messaging Server                   |
|   | Calendar Server                    |
|   | Instant Messaging                  |
|   | Access Manager                     |
|   | Directory Server                   |
|   | Application Server oder Web Server |
| Netzwerkidentitätsdienste                     | Access Manager                     |
|   | Directory Server                   |
|   | Web Server                         |
| Web- und Anwendungsdienste                    | Application Server                 |
|   | Message Queue                      |
|   | Web Server                         |
| Verfügbarkeitsdienste                         | Sun Cluster                        |
|   | Sun Cluster Agents                 |

Weitere Informationen zu den Java Enterprise System-Diensten und -Komponenten sowie zu den Architekturkonzepten von Java Enterprise System finden Sie im Handbuch *Sun Java Enterprise System 2005Q4 Technischer Überblick*.

## Integrierte Dienste und benutzerdefinierte Dienste

Auf Java Enterprise System basierte Bereitstellungslösungen lassen sich in der Regel in zwei allgemeine Kategorien einteilen:

- **80:20-Bereitstellungen.** Diese Lösungen bestehen hauptsächlich aus Diensten, die von Java Enterprise System bereitgestellt werden. Java Enterprise System stellt etwa 80 % oder mehr der Dienste bereit.
- **20:80-Bereitstellungen.** Diese Lösungen bestehen aus zahlreichen benutzerdefinierten Diensten und Drittanbieteranwendungen.

Bei den Kategorieren 80:20 und 20:80 handelt es sich um sehr allgemeine Kategorien. Der genaue Prozentsatz des zur Verfügung gestellten Diensttyps spielt dabei keine Rolle. Der Prozentsatz weist jedoch auf den Umfang der Anpassungen hin, die in einer Lösung enthalten sind.

Java Enterprise System ist aufgrund der umfangreichen Gruppe der von Java ES bereitgestellten Dienste optimal für 80:20-Bereitstellungen geeignet. Es ist beispielsweise relativ einfach, ein unternehmensweites Kommunikationssystem oder ein unternehmensweites Portalsystem mithilfe der Dienste von Java Enterprise System bereitzustellen.

Für Bereitstellungen, die eine benutzerdefinierte Entwicklung erfordern, unterstützt Java Enterprise System die Erstellung und Integration benutzerdefinierter Dienste und Anwendungen.

Die meisten der in der Tabelle „Systemdienste“ auf Seite 17 aufgelisteten Dienstkategorien können für die Verteilung von 80:20-Bereitstellungen verwendet werden. Kommunikations- und Zusammenarbeitsdienste stellen den Endbenutzern beispielsweise E-Mail-, Kalender- und Instant Messaging-Funktionen zur Verfügung, mit denen die Benutzer Inhalte zusammenfassen und anpassen können. Genauso ermöglichen Ihnen die Dienstkategorien für die Netzwerkidentität und das Unternehmensportal die Installation und Konfiguration unternehmensweiter Anwendungen, ohne dass Sie benutzerdefinierte Dienste entwickeln oder integrieren müssen.

Für Unternehmenslösungen, bei denen eine benutzerdefinierte Entwicklung von J2EE-Plattformdiensten erforderlich ist, kann Application Server, Message Queue oder Web Server genutzt werden, die über die Web- und Anwendungsdienste von Java Enterprise System bereitgestellt werden.

Unternehmensbereitstellungen können hinsichtlich der Anzahl der erforderlichen benutzerdefinierten Dienste erheblich variieren. Aufgrund der Interoperabilität zwischen den Java Enterprise System-Diensten können Sie eine eigene Dienste-Suite erstellen, die den speziellen Anforderungen Ihres Unternehmens gerecht wird.

## Migration auf Java Enterprise System

Die Planung, der Entwurf und die Implementierung einer Unternehmenslösung, bei der Java Enterprise System verwendet wird, hängen im Wesentlichen von Ihrer aktuellen Bereitstellungsstrategie ab. In Unternehmen, die zum ersten Mal eine Bereitstellungsplanung planen, werden Planung, Entwurf und Implementierung hauptsächlich von den speziellen Anforderungen des Unternehmens bestimmt. Lösungen für die erstmalige Bereitstellung sind jedoch nicht die Regel. Java Enterprise System wird typischerweise dazu verwendet, um vorhandene Unternehmenslösungen zu verbessern oder um ältere Versionen von Java Enterprise System-Komponenten zu aktualisieren.

Wenn Sie vorhandene Lösungen ersetzen oder aktualisieren, müssen Sie zusätzliche Schritte bei der Planung, beim Entwurf und bei der Implementierung ausführen, um sicherzustellen, dass die vorhandenen Daten beibehalten und die Software richtig auf die aktuellen Versionen aktualisiert wird. Wenn Sie die Analyse durchführen und die in diesem Handbuch beschriebene Struktur entwerfen, sollten Sie die Vorbereitung und Planung beachten, die erforderlich ist, um vorhandene Softwaresysteme zu ersetzen oder zu aktualisieren.

Weitere Informationen zum Aktualisieren der aktuellen Version von Java Enterprise System und zu Migrationsstrategien für andere Anwendungen erhalten Sie im *Java Enterprise System Aufrüstungs- und Migrationshandbuch*.

---

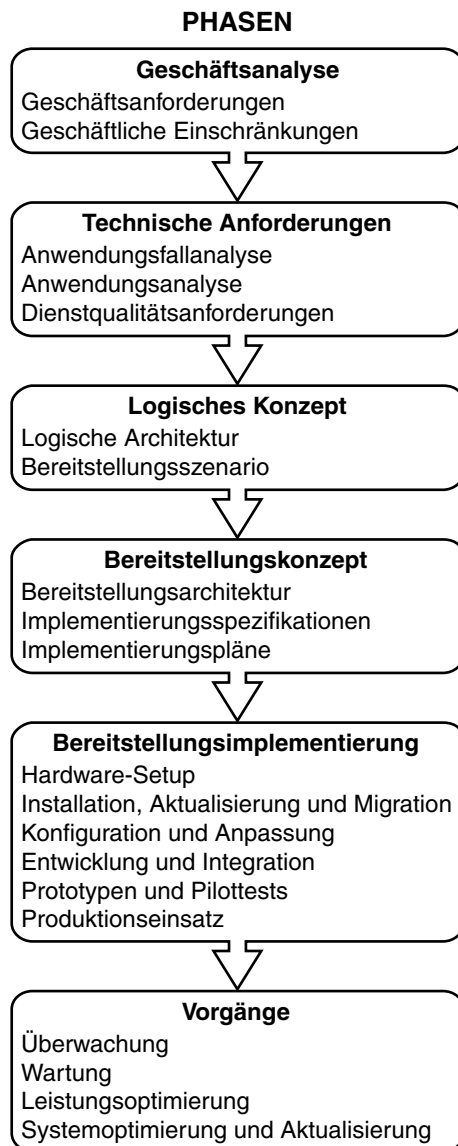
## Informationen zur Bereitstellungsplanung

Die Bereitstellungsplanung ist ein entscheidender Schritt bei der erfolgreichen Implementierung einer Java Enterprise System-Lösung. Jedes Unternehmen verfügt über eigene Ziele, Anforderungen und Prioritäten, die es zu berücksichtigen gilt. Die erfolgreiche Planung beginnt mit der Analyse der Ziele eines Unternehmens und mit der Bestimmung der zum Erreichen der Ziele erforderlichen Geschäftsanforderungen. Die Geschäftsanforderungen müssen anschließend in technische Anforderungen übertragen werden, die als Grundlage für den Entwurf und die Implementierung eines Systems dienen können, das die Ziele des Unternehmens erfüllt.

Eine erfolgreiche Bereitstellungsplanung ist das Ergebnis einer sorgfältigen Vorbereitung, Analyse und eines gelungenen Entwurfs. Fehler und Irrtümer, die in den unterschiedlichen Schritten des Planungsprozesses auftreten, können zu verschiedenartigen Systemausfällen führen. Ein schlecht geplantes System kann schwerwiegende Fehler verursachen. So könnten beispielsweise die Systemleistung nicht ausreichen, die Wartung schwierig, der Betrieb zu teuer und Ressourcen verschwendet werden oder die Skalierbarkeit reicht nicht aus, um steigenden Anforderungen gerecht zu werden.

## Lebenszyklus einer Lösung

Der in der folgenden Abbildung dargestellte Lebenszyklus einer Lösung gibt die Schritte der Planung, des Entwurfs und der Implementierung einer auf Java Enterprise System basierenden Unternehmenssoftwarelösung wieder. Der Lebenszyklus ist hilfreich, um die Phasen eines Bereitstellungsprojekts mitzuverfolgen.



**ABBILDUNG 1-1** Lebenszyklus einer Lösung

Der Lebenszyklus besteht aus aufeinander folgenden Phasen in einer bestimmten Reihenfolge. Jede Phase umfasst entsprechende Aufgaben, deren Ergebnisse jeweils als Ausgangspunkt für die darauf folgenden Phasen dienen. Die Aufgaben innerhalb der einzelnen Phasen sind von Wiederholungen geprägt und erfordern sorgfältige Maßnahmen hinsichtlich Analyse und Konzept, bevor entsprechende Ergebnisse für

die jeweilige Phase generiert werden können. Auch in den frühen Phasen können Wiederholungen auftreten. Während der Phase des Bereitstellungskonzepts könnten Sie beispielsweise feststellen, dass die Analyse in einer früheren Phase nicht ausreichend ist und sorgfältiger ausgeführt werden muss.

In den nachfolgenden Abschnitten dieses Kapitels werden die einzelnen Phasen des Lebenszyklus kurz beschrieben.

## Phase der Geschäftsanalyse

Während der Geschäftsanalyse definieren Sie die Geschäftsziele eines Bereitstellungsprojekts und ermitteln die Geschäftsanforderungen, die für das Erreichen dieser Ziele erforderlich sind. Beachten Sie bei der Ermittlung der Geschäftsanforderungen alle Geschäftseinschränkungen, die ein Hindernis für das Erreichen des Geschäftsziels darstellen könnten. Während des Lebenszyklus bewerten Sie den Erfolg Ihrer Bereitstellungsplanung und schließlich Ihrer Bereitstellungslösung anhand der Analyse, die Sie in der Phase der Geschäftsanalyse durchgeführt haben.

Während der Geschäftsanalysephase erstellen Sie Dokumente mit Geschäftsanforderungen, die Sie später als Ausgangspunkt für die Phase der technischen Anforderungen verwenden.

Weitere Informationen zur Phase der Geschäftsanalyse erhalten Sie in [Kapitel 2](#)

## Phase der technischen Anforderungen

Die Phase der technischen Anforderungen beginnt mit den in der Geschäftsanalyse definierten Geschäftsanforderungen und Geschäftseinschränkungen. Diese werden anschließend in technische Daten übertragen, die für den nachfolgenden Entwurf der Bereitstellungsarchitektur verwendet werden. In den technischen Anforderungen werden die Dienstqualitätsmerkmale (Quality of Service, QoS), wie Leistung, Verfügbarkeit, Sicherheit usw., angegeben.

Während der Phase der technischen Anforderungen erstellen Sie Dokumente, die folgende Informationen enthalten:

- Analyse der Benutzeraufgaben und der Anwendungsmuster.
- Anwendungsfälle, in denen die Benutzerinteraktion mit dem geplanten System simuliert wird.
- Die anhand der Geschäftsanforderungen abgeleiteten Dienstqualitätsanforderungen, bei denen nach Möglichkeit auch die Analyse der Benutzeraufgaben und der Anwendungsmuster berücksichtigt werden sollten.

Die daraus resultierenden Dokumente mit Anwendungsanalyse, Anwendungsfällen und QoS-Anforderungen dienen als Ausgangspunkt für die Phase des logischen Konzepts im Lebenszyklus der Lösung. Die Anwendungsanalyse spielt ebenfalls eine entscheidende Rolle in der Phase des Bereitstellungskonzepts.

Während der Phase der technischen Anforderungen können Sie zudem die Anforderungen für die Dienstebenen, die die Grundlage für die darauffolgende Erstellung der Vereinbarungen auf Dienstebene (Service Level Agreement, SLA) bilden, festlegen. In einer Vereinbarung auf Dienstebene werden die Bedingungen angegeben, unter denen die Kundenunterstützung zur Verfügung gestellt werden muss, damit das System ordnungsgemäß gewartet werden kann. Sie sind im Allgemeinen Bestandteil der Projektgenehmigung in der Phase des Bereitstellungskonzepts.

Weitere Informationen zu den technischen Anforderungen erhalten Sie in [Kapitel 3](#).

## Phase des logischen Konzepts

Während der Erstellung des logischen Konzepts, in der die Anwendungsfälle aus der Phase der technischen Anforderungen als Ausgangspunkt dienen, bestimmen Sie die Java Enterprise System-Komponenten, die für die Implementierung einer Lösung erforderlich sind. Sie ermitteln die Komponenten, die diese Java ES-Komponenten unterstützen und bestimmen zusätzliche benutzerdefinierte Komponenten, die zum Erfüllen der Geschäftsanforderungen erforderlich sind. Anschließend ordnen Sie die Komponenten innerhalb einer logischen Architektur zu, in der die gegenseitigen Abhängigkeiten der Komponenten dargestellt sind. In der logischen Architektur ist keine Hardware angegeben, die für die Implementierung der Lösung erforderlich ist.

Das Ergebnis der Phase des logischen Konzepts ist die logische Architektur. Die logische Architektur selbst reicht nicht aus, um mit der Erstellung des Bereitstellungskonzepts zu beginnen. Sie benötigen ebenfalls die QoS-Anforderungen aus der Phase der technischen Anforderungen. Die logische Architektur und die QoS-Anforderungen aus der Phase der technischen Anforderungen bilden ein Bereitstellungsszenario. Das Bereitstellungsszenario wiederum dient als Ausgangspunkt für die Phase des Bereitstellungskonzepts.

Weitere Informationen zum logischen Konzept erhalten Sie in [Kapitel 4](#).

## Phase des Bereitstellungskonzepts

In der Phase des Bereitstellungskonzepts ordnen Sie die in der logischen Architektur angegebenen Komponenten einer physischen Umgebung zu und erstellen so eine Bereitstellungsarchitektur auf übergeordneter Ebene. Zudem erstellen Sie eine Implementierungsspezifikation, in der die detaillierten Angaben der unteren Ebene für die Erstellung der Bereitstellungsarchitektur enthalten sind. Darüber hinaus erstellen Sie eine Reihe von Plänen und Spezifikationen mit detaillierten Informationen zu unterschiedlichen Aspekten der Implementierung der Softwarelösung.

Die Projektgenehmigung findet in der Bereitstellungskonzeptphase statt. Bei der Projektgenehmigung werden die Kosten der Bereitstellung bewertet. Wenn die Genehmigung erfolgt ist, werden die Verträge für die Implementierung der



Bereitstellung unterschrieben und die Ressourcen zur Erstellung des Projekts werden erworben. Häufig findet die Projektgenehmigung nach der Angabe der Implementierungsspezifikationen statt. Die Genehmigung kann jedoch auch nach Abschluss der Bereitstellungsarchitektur erfolgen.

Die Ergebnisse der Bereitstellungs-konzeptphase beinhalten Folgendes:

- **Bereitstellungsarchitektur.** Ein Entwurfsdokument der oberen Ebene, in dem die Zuordnung der Komponenten zur Hardware und Software des Netzwerks angegeben ist.
- **Implementierungsspezifikationen.** Detaillierte Spezifikationen, die zur Erstellung der Bereitstellung herangezogen werden.
- **Implementierungspläne.** Eine Reihe von Plänen und Spezifikationen, die unterschiedliche Aspekte der Implementierung einer Unternehmenssoftwarelösung abdecken. Zu den Implementierungsplänen gehören ein Migrationsplan, ein Installationsplan, ein Benutzerverwaltungsplan, ein Testplan und sonstige Pläne.

Weitere Informationen zum Bereitstellungs-konzept erhalten Sie in [Kapitel 5](#).

## Implementierungsphase

In der Implementierungsphase arbeiten Sie mit Spezifikationen und Plänen, die Sie in der Phase des Bereitstellungs-konzepts erstellt haben, um die Bereitstellungsarchitektur zu entwickeln und die Lösung zu implementieren. Je nach Art Ihres Bereitstellungsprojekts umfasst diese Phase einige oder alle der folgenden Aufgaben:

- Installation und Konfiguration der Hardwareinfrastruktur
- Installation und Konfiguration der Software
- Angabe von Benutzern und Ressourcen innerhalb einer LDAP-Verzeichnisstruktur
- Migration von Daten von vorhandenen Verzeichnissen und Datenbanken entsprechend einem Benutzerverwaltungsplan
- Erstellung und Implementierung einer Pilot- und einer Prototypbereitstellung in einer Testumgebung
- Entwurf und Ausführung eines Funktionstests zur Bestimmung der Kompatibilität mit den Systemanforderungen
- Entwurf und Ausführung von Belastungstests zur Bestimmung der Leistung unter maximaler Auslastung
- Entwicklung und Integration von benutzerdefinierten Unternehmensanwendungen
- Erstellen einer Produktionsbereitstellung, die phasenweise in die Produktion integriert werden kann

Sobald sich eine Bereitstellung in Produktion befindet, gehen Sie in die Betriebsphase des Lösungslebenszyklus über.

Weitere Informationen zur Implementierungsphase erhalten Sie in [Kapitel 6](#).

## Betriebsphase

Die Betriebsphase umfasst die Aufgaben, die erforderlich sind, um dafür zu sorgen, dass die Implementierung der Bereitstellung problemlos läuft. Diese Phase umfasst Folgendes:

- Überwachung der Bereitstellung, um sicherzustellen, dass das System nach Plan läuft
- Leistungsoptimierung, um zu gewährleisten, dass die bereitgestellte Software optimal läuft
- Bereitstellung geplanter Wartungsmaßnahmen für einen problemlosen Betrieb und nicht geplanter Wartungen nach Bedarf
- Aktualisierung der Software und Hardware bei Bedarf

Detaillierte Informationen zur Betriebsphase sind nicht Bestandteil dieses Handbuchs.

## Geschäftsanalyse

---

Während der Phase der Geschäftsanalyse innerhalb des Lebenszyklus einer Lösung definieren Sie die Geschäftsziele, indem Sie ein Geschäftsproblem analysieren und die Geschäftsanforderungen und -einschränkungen identifizieren, mit denen dieses Ziel erreicht wird.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Informationen zur Geschäftsanalyse“ auf Seite 27
- „Definieren der Geschäftsanforderungen“ auf Seite 28
- „Definieren der geschäftlichen Einschränkungen“ auf Seite 33

---

## Informationen zur Geschäftsanalyse

Die Geschäftsanalyse beginnt mit der Festlegung der Geschäftsziele. Anschließend analysieren Sie die zu lösenden Geschäftsprobleme und identifizieren die Geschäftsanforderungen, die für das Erreichen dieser Ziele erfüllt werden müssen. Berücksichtigen Sie auch sämtliche geschäftliche Einschränkungen, die ein Erreichen dieser Ziele beeinträchtigen. Das Ergebnis der Analyse der Geschäftsanforderungen und -einschränkungen ist eine Gruppe von Dokumenten mit Geschäftsanforderungen.

Diese Dokumente hinsichtlich der Geschäftsanforderungen verwenden Sie als Grundlage für die Ableitung technischer Anforderungen in der Phase der technischen Anforderungen. Während des Lebenszyklus einer Lösung bewerten Sie den Erfolg Ihrer Bereitstellungsplanung und schließlich Ihrer Lösung anhand der Analyse, die Sie in der Phase der Geschäftsanalyse durchgeführt haben.

---

## Definieren der Geschäftsanforderungen

Eine einfache Formel zum Bestimmen sämtlicher Geschäftsanforderungen gibt es nicht. Sie bestimmen die Anforderungen basierend auf der Zusammenarbeit mit den Interessengruppen, die eine Softwarelösung benötigen, Ihrer Kenntnis der Geschäftsdomäne und dem angewandten kreativen Denken.

In diesem Abschnitt werden einige Faktoren behandelt, die bei der Definition der Geschäftsanforderungen berücksichtigt werden müssen.

## Festlegen der Geschäftsziele

Die Geschäftsanalyse sollte die Ziele eines Bereitstellungsprojekts deutlich machen. Durch die Festlegung klarer Ziele können Konzeptentscheidungen genauer ausgerichtet werden und das Projekt bleibt auf dem richtigen Weg. Die Gegenüberstellung der Geschäftsziele und der aktuellen Vorgänge hilft zudem, Konzeptentscheidungen zu treffen.

## Bereich

Die Geschäftsanforderungen sollten den Geltungsbereich des Bereitstellungsprojekts klar abstecken. Stellen Sie sicher, dass Sie Bereiche bestimmen, die gelöst werden können, und vermeiden Sie es, unbegrenzte Anforderungen festzulegen, die das Ziel entweder unklar oder unerreichbar machen. Ein schlecht definierter Bereich kann zu einem Bereitstellungskonzept führen, das die Bedürfnisse des Unternehmens nicht ausreichend einbezieht oder das übermäßig viele Ressourcen erfordert.

## Prioritäten

Setzen Sie Prioritäten in Bezug auf Ihre Ziele, um sicherzustellen, dass die wichtigsten Aspekte der Bereitstellung zuerst erreicht werden. Wenn Sie nur begrenzte Ressourcen zur Verfügung haben, müssen einige Ziele eventuell verschoben oder abgewandelt werden. Umfangreiche und komplexe Bereitstellungen erfordern beispielsweise eine phasenweise Implementierung der Lösung. Durch die Festlegung von Prioritäten geben Sie Richtlinien für Entscheidungen an, die möglicherweise getroffen werden müssen, damit Ihr Bereitstellungskonzept von den Interessengruppen akzeptiert wird.

## Entscheidende Qualitäten

Legen Sie Bereiche fest, die für den Erfolg entscheidend sind, damit die Interessengruppen und Entwickler sich auf die wichtigsten Kriterien konzentrieren können.

## Wachstumsfaktoren

Berücksichtigen Sie beim Festlegen von Geschäftszielen nicht nur die aktuellen Bedürfnisse des Unternehmens, sondern versuchen Sie vorherzusagen, inwiefern sich diese Bedürfnisse über längere Zeiträume ändern und entwickeln könnten. Sie wollen schließlich keine Lösung, die schon vorzeitig veraltet ist.

## Sicherheitsspielraum

Das Konzept Ihrer Lösung wird anhand von Annahmen entwickelt, die während der Geschäftsanalysephase aufgestellt werden. Diese Annahmen können aus verschiedenen Gründen, wie beispielsweise unzureichende Daten, Fehler bei der Bewertung oder unvorhergesehene externe Ereignisse, ungenau sein. Planen Sie deshalb unbedingt einen Sicherheitsspielraum ein, nicht nur in Bezug auf Ihre Geschäftsziele, sondern während der gesamten Planung, damit die Lösung, die Sie entwickeln, auch unvorhersehbare Ereignisse bewältigen kann.

## Kenntnis der Benutzerbedürfnisse

Bringen Sie in Erfahrung, an welche Benutzerzielgruppen sich die Lösung richtet, welche Bedürfnisse sie haben und welche Vorteile für sie zu erwarten sind. Folgende Liste stellt beispielsweise eine Möglichkeit für die Kategorisierung von Benutzern dar:

- Nur derzeitige Mitarbeiter
- Derzeitige und frühere Mitarbeiter
- Administratoren
- Aktive Kunden
- Alle Kunden
- Mitgliederstandort
- Allgemeinheit
- Eingeschränkter Zugriff

Wenn Sie die erwarteten Vorteile für die Benutzer eindeutig festlegen, können Konzeptentscheidungen leichter getroffen werden. Im Folgenden werden einige Beispiele als Vorteile aufgeführt, die Benutzern durch eine Lösung entstehen können:

- Fernzugriff auf Unternehmensressourcen
- Unternehmensweite Zusammenarbeit
- Vereinfachung täglicher Aufgaben
- Gemeinsame Nutzung von Ressourcen durch räumlich voneinander getrennte Teams
- Steigerung der Produktivität
- Eigenadministration durch Endbenutzer

## Erarbeiten von Betriebsanforderungen

Formulieren Sie Betriebsanforderungen als eine Reihe von funktionellen Anforderungen mit klar definierten Zielen. Normalerweise erstellen Sie Betriebsspezifikationen für folgende Bereiche:

- Endbenutzerfunktionalität
- Reduzierte Antwortzeit
- Verfügbarkeit und Betriebszeit
- Reduzierte Fehlerrate
- Archivierung und Aufbewahrung von Informationen

Formulieren Sie die Betriebsanforderungen in messbaren Begriffen, die alle Interessengruppen verstehen. Vermeiden Sie unklare Ausdrücke, wie beispielsweise „adäquate Endbenutzerantwortzeit“. Beispiel für mögliche Betriebsanforderungen:

- Fähigkeit, Dienste innerhalb von 10 Minuten nach einem Ausfall wiederherzustellen.
- Fähigkeit, die eingehenden Nachrichten der letzten 48 Stunden wiederzugeben.
- Abschluss von Online-Transaktionen in Spitzenzeiten in maximal 60 Sekunden.
- Endbenutzer-Authentifizierung in Spitzenzeiten innerhalb von vier Sekunden.

## Unterstützung vorhandener Anwendungsmuster

Formulieren Sie vorhandene Anwendungsmuster als klar messbare Ziele. Folgende Fragen helfen bei der Ermittlung derartiger Ziele.

- Wie werden die aktuellen Dienste genutzt?
- Welche Anwendungsmuster sind vorhanden (z. B. sporadische, häufige oder erhebliche Benutzerauslastung)?
- Welche Sites rufen Ihre Benutzer normalerweise auf?
- Nachrichten welcher Größe werden normalerweise von Ihren Benutzern gesendet?
- Wie viele Transaktionen führen die Benutzer normalerweise an einem Tag oder in einer Stunde aus?

Beobachten Sie die Benutzer, die auf Ihre Dienste zugreifen. Kenntnisse darüber, wann und für wie lange die Benutzer auf die vorhandenen Dienste zugreifen, sind Schlüsselfaktoren bei der Ermittlung Ihrer Ziele. Wenn diese Muster nicht durch unternehmenseigene Erfahrungen gebildet werden können, ziehen Sie die Erfahrungen ähnlicher Unternehmen zurate.

## Kenntnis der Unternehmenskultur

Bei der Anforderungsanalyse sollten verschiedene Aspekte der Unternehmenskultur und -politik berücksichtigt werden. Wenn die Unternehmenskultur nicht ausreichend beachtet wird, wird die Lösung möglicherweise nicht gut aufgenommen oder kann nur schwer implementiert werden.

## Interessengruppen

Finden Sie heraus, welche Personen und Organisationen stark am Erfolg der geplanten Lösung interessiert sind. Alle Interessengruppen sollten aktiv an der Definition der Geschäftsziele und -anforderungen beteiligt sein. Wenn eine Interessengruppe nicht beteiligt ist oder nicht über geplante Änderungen informiert wird, kann die Planung erheblich beeinträchtigt werden. Eine derartige Interessengruppe könnte die Implementierung der Bereitstellung sogar blockieren.

## Standards und Richtlinien

Stellen Sie sicher, dass Sie die Standards und Richtlinien des Unternehmens kennen, für das die Lösung vorgesehen ist. Diese Standards und Richtlinien wirken sich möglicherweise auf die technischen Aspekte des Konzepts, die Produktauswahl und die Bereitstellungsmethode aus.

Ein Beispiel stellt die Vertraulichkeit der Personaldaten dar, die sich bei der Personalabteilung oder beim Bereichsleiter befinden, von denen sie auch kontrolliert werden. Ein anderes Beispiel stellen die Vorgehensweisen des Unternehmens in Bezug auf die Änderungsverwaltung dar. Änderungsverwaltungsrichtlinien können sich erheblich auf die Akzeptanz einer Lösung auswirken und die Implementierungsmethode und den Zeitplan beeinflussen.

## Regulierungsanforderungen

Regulierungsanforderungen hängen in hohem Maße von der Art des Unternehmens ab. Untersuchen Sie sämtliche Regulierungsanforderungen, die sich auf die Bereitstellung auswirken können. Viele Unternehmen und Behörden schreiben die Einhaltung von Zugänglichkeitsstandards vor. Bei der Bereitstellung globaler Lösungen müssen die im Ausland geltenden Gesetze und Regulierungen beachtet werden. In vielen europäischen Ländern herrschen beispielsweise strikte Regeln für das Speichern persönlicher Daten.

## Sicherheit

Einige Ziele, die Sie ermitteln, bringen möglicherweise Sicherheitsprobleme mit sich, die besondere Berücksichtigung erfordern. Heben Sie bestimmte Sicherheitsziele hervor, die für die Lösung besonders wichtig sind. Beispiel:

- Autorisierter Zugriff auf vertrauliche Informationen
- Rollenbasierter Zugriff auf vertrauliche Informationen
- Sichere Kommunikation zwischen entfernten Standorten
- Aufrufen von Remote-Anwendungen auf lokalen Systemen
- Sichere Transaktionen mit anderen Unternehmen und Organisationen
- Durchsetzung der Sicherheitsrichtlinien

## Standortverteilung

Die geografische Verteilung der Standorte und die Bandbreite zwischen den Standorten spielt eine Rolle für die Konzeptentscheidungen. Zudem erfordern einige Standorte möglicherweise eine lokale Verwaltung.

Derartige geografische Überlegungen können sich auf die Schulungskosten für das Projekt, die Komplexität usw. auswirken. Geben Sie Anforderungen, die sich aus der geografischen Standortverteilung ergeben, klar an. Heben Sie hervor, welche Standorte für den Erfolg des Konzepts entscheidend sind.

## Schrittweise Annäherung

Häufig betrachtet man eine Softwarelösung als komplettes umfassendes System. Die Bereitstellung des kompletten Systems erfolgt jedoch oft als schrittweise Annäherung.

Bei einer schrittweisen Annäherung entwerfen Sie normalerweise einen Fahrplan, der Meilensteine aufweist, die dann zur endgültigen umfassenden Lösung führen. Zusätzlich müssen Sie möglicherweise kurzfristige Pläne für Teilaspekte der umfassenden Lösung entwerfen, deren Implementierung erst zu einem späteren Zeitpunkt vorgesehen ist.

Die schrittweise Annäherung hat folgende Vorteile:

- Sie können Anpassungen an Anforderungsänderungen vornehmen, die durch das Wachstum des Unternehmens entstehen.
- Sie können die vorhandene Infrastruktur als Übergang zu Ihrer endgültigen Bereitstellungsimplementierung nutzen.
- Sie können Anforderungen in Bezug auf die Investitionsausgaben einbeziehen.
- Sie können wenig Personal optimal nutzen.
- Sie können Voraussetzungen für Partnerschaften schaffen.

## Kenntnis der Vereinbarungen auf Dienstebene

In einer Vereinbarung auf Dienstebene (Service Level Agreement, SLA) werden die leistungsbezogenen Mindestanforderungen angegeben. Wenn diese Anforderungen nicht erfüllt werden, geht hieraus ebenfalls hervor, auf welcher Ebene und in welchem Umfang Kundenunterstützung bereitgestellt werden muss. Eine Vereinbarung auf Dienstebene basiert auf den Geschäftsanforderungen, die bei der Geschäftsanalyse definiert werden. Später, in der Phase der technischen Anforderungen, werden diese Anforderungen als auf die Dienstebene bezogene Anforderungen angegeben. Die SLA wird bei der Projektgenehmigung (die in der Bereitstellungsphase erfolgt) unterzeichnet.

Sie sollten eine SLA entwickeln, die Bereiche wie Betriebszeit, Antwortzeit, Nachrichtenübermittlungszeit und Wiederherstellung beinhaltet. Eine SLA sollte einen Überblick über das System enthalten und die Rollen und Verantwortlichkeiten von



Support-Unternehmen klären. Außerdem sollte darin geregelt sein, wie Dienstebenen, Änderungsanforderungen usw. bemessen werden. Bei der Ermittlung der Reichweite einer SLA ist es entscheidend, die Erwartungen des Unternehmens in Bezug auf die Systemverfügbarkeit zu bestimmen.

---

## Definieren der geschäftlichen Einschränkungen

Geschäftliche Einschränkungen spielen eine bedeutende Rolle, wenn die Art eines Bereitstellungsprojekts bestimmt wird. Ein Schlüssel für ein erfolgreiches Bereitstellungskonzept ist die Ermittlung einer optimalen Methode, mit der Geschäftsanforderungen innerhalb der bekannten geschäftlichen Einschränkungen erfüllt werden. Bei den geschäftlichen Einschränkungen kann es sich um finanzielle Einschränkungen, physikalische Einschränkungen (z. B. die Netzwerkkapazität), zeitliche Einschränkungen (z. B. die Fertigstellung vor bedeutenden Ereignissen, wie etwa der nächsten Jahresversammlung) oder jegliche andere Einschränkungen, die sich auf das Erreichen des Geschäftsziels auswirken könnten, handeln.

In diesem Abschnitt werden einige Faktoren beschrieben, die bei der Definition der geschäftlichen Einschränkungen berücksichtigt werden müssen.

## Migrationsprobleme

Normalerweise ersetzt oder ergänzt ein Bereitstellungsprojekt vorhandene Softwareinfrastrukturen und -daten. Eine neue Lösung muss in der Lage sein, Daten und Vorgehensweisen aus der vorhandenen Infrastruktur in die neue Lösung zu migrieren, wobei oft die Interoperabilität mit vorhandenen Anwendungen bewahrt wird. Eine Analyse der aktuellen Infrastruktur ist erforderlich, um zu ermitteln, in welchem Umfang Migrationsprobleme bei der geplanten Lösung eine Rolle spielen.

## Planungsmandate

Der Plan für die Implementierung einer Lösung kann sich auf die Konzeptentscheidungen auswirken. Aggressive Pläne führen möglicherweise dazu, dass von Zielen Abstand genommen wird, Prioritäten geändert werden oder eine schrittweise Annäherung an die Lösung vorgenommen wird. Innerhalb eines Plans sind möglicherweise wichtige Meilensteine vorhanden, die ebenso in Betracht gezogen werden sollten. Als Meilensteine können interne Ereignisse, wie beispielsweise die geplante Einführung eines Dienstes, oder externe Ereignisse, wie beispielsweise der Beginn eines neuen Schuljahres, dienen.

## Budgeteinschränkungen

Die meisten Bereitstellungsprojekte unterliegen einem Budget. Berücksichtigt werden die Kosten für die Entwicklung der geplanten Lösung und die für die Wartung der Lösung über eine bestimmte Lebensdauer erforderlichen Ressourcen:

- **Vorhandene Hardware- und Netzwerkinfrastruktur.** Die Abhängigkeit von vorhandener Infrastruktur kann sich auf das Konzept eines Systems auswirken.
- **Für die Implementierung der Lösung erforderliche Entwicklungsressourcen.** Begrenzte Entwicklungsressourcen (Hardware, Software und Personal) machen eine schrittweise Bereitstellung naheliegend. Sie müssen möglicherweise dieselben Ressourcen oder Entwicklungsteams für jede inkrementelle Phase Ihrer Implementierung wiederverwenden.
- **Wartung, Verwaltung und Unterstützung.** Analysieren Sie die für die Verwaltung, Wartung und Unterstützung verfügbaren Ressourcen. Begrenzte Ressourcen wirken sich möglicherweise auf Konzeptentscheidungen aus.

## Gesamtkosten

Neben der Wartung, Verwaltung und Unterstützung müssen weitere Faktoren analysiert werden, die sich auf die Gesamtkosten auswirken. Eventuell notwendige Hardware- und Softwareaufrüstungen, ein gesteigerter Strombedarf durch die Lösung, Telekommunikationskosten und andere Faktoren können sich auf die Auslagen auswirken. Vereinbarungen auf Dienstebene, die Verfügbarkeitsstufen für die Lösung vorschreiben, wirken sich zudem auf die Gesamtkosten aus, da eine erhöhte Redundanz nötig wird.

Die Implementierung einer Lösung sollte auf die Lösung angerechnet eine Rendite erzeugen. Bei der Analyse der Investitionsrentabilität werden normalerweise die finanziellen Vorteile berechnet, die durch die Investitionsausgabe erlangt wurden.

Bei der Einschätzung der finanziellen Vorteile durch eine Lösung müssen die zu erreichenden Ziele sorgfältig im Vergleich zu anderen Möglichkeiten analysiert werden, mit denen diese Ziele erreicht werden, sowie im Vergleich zu den Kosten, wenn keine Maßnahmen ergriffen werden.

## Technische Anforderungen

---

Während der Phase der technischen Anforderungen im Lösungslebenszyklus führen Sie eine Anwendungsanalyse durch, erstellen Anwendungsfälle und bestimmen die Dienstqualitätsanforderungen für die vorgeschlagene Bereitstellungslösung.

Dieses Kapitel enthält folgende Abschnitte:

- „Informationen zu technischen Anforderungen“ auf Seite 35
- „Anwendungsanalyse“ auf Seite 36
- „Anwendungsfälle“ auf Seite 38
- „Dienstqualitätsanforderungen“ auf Seite 39
- „Anforderungen an die Dienstebene“ auf Seite 48

---

## Informationen zu technischen Anforderungen

Die Analyse der technischen Anforderungen basiert auf den Dokumenten zu den Geschäftsanforderungen, die während der Phase der Geschäftsanalyse des Lösungslebenszyklus erstellt wurden. Verwenden Sie die Geschäftsanalyse als Grundlage für die Durchführung folgender Aufgaben:

- Führen Sie eine Anwendungsanalyse durch, mit deren Hilfe Sie die erwarteten Lastbedingungen bestimmen können.
- Erstellen Sie Anwendungsfälle, die typische Benutzerinteraktionen mit dem System beispielhaft darstellen.
- Erstellen Sie einen Satz an Dienstqualitätsanforderungen (Quality of Service Requirements, QoS), die definieren, welche Leistung eine bereitgestellte Lösung in Bereichen wie Antwortzeit, Verfügbarkeit, Sicherheit usw. erbringen muss.

Die Dienstqualitätsanforderungen werden anhand der Anwendungsanalyse und der Anwendungsfälle entwickelt und berücksichtigen die zuvor formulierten Geschäftsanforderungen und -einschränkungen.

Die Geschäftsanforderungen werden später mit logischen Architekturen in der Phase des logischen Konzepts gepaart, um ein Bereitstellungsszenario zu entwerfen. Das Bereitstellungsszenario stellt den wichtigsten Beitrag in der Phase des logischen Konzepts des Lösungslebenszyklus dar.

So wie es auch für die Geschäftsanalyse der Fall ist, existiert für die Analyse der technischen Anforderungen keine Formel, mit der die Anwendungsanalyse, die Anwendungsfälle und die Systemanforderungen einfach erstellt werden könnten. Die Analyse der technischen Anforderungen setzt das Verständnis des jeweiligen Geschäftsbereichs, der Geschäftsziele und der zugrundeliegenden Systemtechnologie voraus.

---

## Anwendungsanalyse

In der Anwendungsanalyse werden die verschiedenen Benutzer der zu entwickelnden Lösung und die Anwendungsmuster für diese Benutzer ermittelt. Die zusammengetragenen Informationen bilden eine Grundlage für eine Schätzung der Lastbedingungen des Systems. Die Informationen aus der Anwendungsanalyse sind außerdem bei der Gewichtung der Anwendungsfälle, wie unter „Anwendungsfälle“ auf Seite 38 beschrieben, hilfreich.

Bei der Anwendungsanalyse sollten Sie Benutzer wann immer möglich mit einbeziehen und befragen, vorhandene Daten zu Anwendungsmustern analysieren und die Entwickler und Administratoren vorheriger Systeme befragen. In der folgenden Tabelle werden die Faktoren aufgelistet, die bei der Durchführung einer Anwendungsanalyse zu berücksichtigen sind.

**TABELLE 3–1** Faktoren der Anwendungsanalyse

| Thema                          | Beschreibung  |
|--------------------------------|---|
| Benutzeranzahl und Benutzertyp | <p>Ermitteln Sie, wie viele Benutzer Ihre Lösung unterstützen muss und kategorisieren Sie die Benutzer gegebenenfalls.</p> <p>Beispiel:</p> <ul style="list-style-type: none"> <li>■ Eine Business-to-Customer (B2C)-Lösung muss möglicherweise für eine große Besucherzahl ausgelegt sein, jedoch nur für eine geringe Anzahl an Benutzern, die sich registrieren und Geschäftstransaktionen vornehmen.</li> <li>■ Eine Business-to-Employee (B2E)-Lösung muss üblicherweise für alle Mitarbeiter ausgelegt sein, selbst wenn einige Mitarbeiter möglicherweise Zugriff von außerhalb des gemeinsamen Netzwerks benötigen. In einer B2E-Lösung benötigen Manager möglicherweise Berechtigungen für Bereiche, auf die ein normaler Mitarbeiter keinen Zugriff hat.</li> </ul> |
| Aktive und inaktive Benutzer   | <p>Ermitteln Sie die Anwendungsmuster und das Anwendungsverhältnis von aktiven zu inaktiven Benutzern.</p> <p>Aktive Benutzer sind am System angemeldete Benutzer und interagieren mit den Systemdiensten. Inaktive Benutzer können Benutzer sein, die nicht angemeldet sind, Benutzer, die angemeldet sind, aber nicht mit den Systemkomponenten interagieren oder Benutzer, die sich in der Datenbank befinden, sich jedoch nie anmelden.</p>   |
| Verwaltungsb Benutzer          | <p>Ermitteln Sie die Benutzer, die auf das bereitgestellte System zugreifen können, um die Bereitstellung zu überwachen, zu aktualisieren und zu unterstützen.</p> <p>Ermitteln Sie sämtliche verwaltungsspezifische Anwendungsmuster, die die technischen Anforderungen beeinflussen könnten (z. B. die Verwaltung der Bereitstellung von außerhalb der Firewall aus).</p>   |
| Anwendungsmuster               | <p>Ermitteln Sie, auf welche Weise die verschiedenen Benutzertypen auf das System zugreifen und formulieren Sie Ziele für die erwartete Auslastung.</p> <p>Beispiel:</p> <ul style="list-style-type: none"> <li>■ Gibt es Spitzenzeiten, zu denen die Auslastung besonders hoch ist?</li> <li>■ Wie sind die normalen Geschäftszeiten?</li> <li>■ Sind die Benutzer weltweit verteilt?</li> <li>■ Wie hoch ist die erwartete Dauer der Benutzerverbindungen?</li> </ul>   |

**TABELLE 3-1** Faktoren der Anwendungsanalyse (Fortsetzung)

| Thema                           | Beschreibung   |
|---------------------------------|--|
| Anstieg der Benutzerzahl        | <p>Ermitteln Sie, ob die Größe des Benutzerstamms feststehend ist oder ob für die Bereitstellung ein Anstieg der Benutzerzahl erwartet wird.</p> <p>Wird ein Anstieg des Benutzerstamms erwartet, versuchen Sie, realistische Prognosen des Anstiegs zu erstellen.</p>   |
| Benutzertransaktionen           | <p>Ermitteln Sie den Typ der Benutzertransaktionen, die unterstützt werden müssen. Diese Benutzertransaktionen können in Anwendungsfälle umgesetzt werden.</p> <p>Beispiel:</p> <ul style="list-style-type: none"><li>■ Welche Aufgaben führen die Benutzer durch?</li><li>■ Bleiben Benutzer nach Ihrer Anmeldung angemeldet? Führen die Benutzer üblicherweise einige Aufgaben durch und melden sich dann wieder ab?</li><li>■ Macht eine weitgehende Zusammenarbeit der Benutzer möglicherweise die Bereitstellung von gemeinsamen Kalendern, Webkonferenzen und internen Webseiten erforderlich?</li></ul> |
| Benutzerstudien und Statistiken | <p>Verwenden Sie bereits vorhandene Benutzerstudien und andere Quellen, um die verschiedenen Verhaltensmuster der Benutzer zu ermitteln.</p> <p>Oft liegen in Firmen oder Industrieunternehmen Benutzerstudien vor, denen Sie nützliche Informationen über die Benutzer entnehmen können. Die Protokolldateien vorhandener Anwendungen enthalten möglicherweise statistische Daten, die für die Erstellung von Prognosen für das System hilfreich sein können.</p>   |

---

## Anwendungsfälle

Anwendungsfälle stellen ein Modell für die typische Benutzerinteraktion mit der zu entwickelnden Lösung dar und beschreiben den vollständigen Verlauf eines Vorgangs aus der Perspektive eines Endbenutzers. Indem das Konzept vorrangig auf der Grundlage vollständiger Anwendungsfälle entwickelt wird, wird sichergestellt, dass der Fokus kontinuierlich auf der Entwicklung einer Lösung mit der erwarteten Funktionalität liegt. Anwendungsfälle stellen die wichtigste Grundlage eines logischen Konzepts dar.

Weisen Sie den Anwendungsfällen eine relative Gewichtung zu, wobei die höchste Gewichtung die Fälle erhalten, die für die häufigsten Benutzeraufgaben stehen. Die Gewichtung von Anwendungsfällen ermöglicht Ihnen die Fokussierung Ihrer Konzeptentscheidungen auf die Systemdienste, die am häufigsten verwendet werden.

Anwendungsfälle können auf zwei Ebenen beschrieben werden.

- **Anwendungsfallberichte.** Beschreibung der einzelnen Anwendungsfälle, einschließlich primärer und alternativer Ereignisabläufe.
- **Anwendungsfalldiagramme.** Diagramme zeigen das Verhältnis zwischen den Agierenden und den Anwendungsfällen und bieten eine eher formale Anordnung der Ereignisabläufe. Diagramme zu Anwendungsfällen sind für die Darstellung langer oder komplexer Anwendungsfälle hilfreich. Für die Erstellung der Anwendungsfalldiagramme verwenden Sie üblicherweise Unified Modeling Language (UML)-Standards.

---

## Dienstqualitätsanforderungen

Dienstqualitätsanforderungen (QoS-Anforderungen) sind technische Spezifikationen, mit der die Systemqualität von Funktionen wie Leistung, Verfügbarkeit, Skalierbarkeit und Wartungseignung bestimmt wird. Die QoS-Anforderungen werden durch die in den Geschäftsanforderungen festgelegten Geschäftsbedürfnisse bestimmt. Wenn Dienste beispielsweise ganzjährig 24 Stunden pro Tag verfügbar sein müssen, muss diese Geschäftsanforderung von der Verfügbarkeitsanforderung erfüllt werden.

Die folgende Tabelle listet die Systemqualitäten auf, die üblicherweise die Grundlage für QoS-Anforderungen bilden.

**TABELLE 3-2** Systemqualitäten mit Auswirkung auf QoS-Anforderungen

| Systemqualität | Beschreibung   |
|----------------|--|
| Leistung       | Die Messung der Antwortzeit und des Datendurchsatzes relativ zu den Benutzerauslastungsbedingungen.  |
| Verfügbarkeit  | Eine Maßangabe, mit der angegeben wird, wie oft Systemressourcen und -dienste für Endbenutzer verfügbar sind. Diese wird häufig als <i>Uptime</i> (verfügbare Betriebszeit) eines Systems bezeichnet.  |
| Skalierbarkeit | Die Möglichkeit, zu einem späteren Zeitpunkt Kapazitäten (und Benutzer) zu einem bereitgestellten System hinzuzufügen. Skalierbarkeit beinhaltet üblicherweise das Hinzufügen von Ressourcen zum System, ohne dass Änderungen an der Bereitstellungsarchitektur vorgenommen werden müssen. |

**TABELLE 3–2** Systemqualitäten mit Auswirkung auf QoS-Anforderungen (Fortsetzung)

| Systemqualität    | Beschreibung  |
|-------------------|---|
| Sicherheit        | Eine komplexe Kombination von Faktoren, die die Integrität eines Systems und seiner Benutzer beschreiben. Zur Sicherheit gehört die Authentifizierung und die Zuweisung von Berechtigungen zu Benutzern sowie Sicherheitsdaten und der sichere Zugriff auf ein bereitgestelltes System. |
| Latente Kapazität | Die Fähigkeit eines Systems, ungewöhnlich hohe Lasten ohne zusätzliche Ressourcen bedienen zu können. Die latente Kapazität spielt eine Rolle in den Verfügbarkeits-, Leistungs- und Skalierbarkeitsanforderungen.  |
| Wartungseignung   | Die Leichtigkeit, mit der ein bereitgestelltes System gewartet werden kann, einschließlich der Systemüberwachung, Problembeseitigung und Aktualisierung von Soft- und Hardwarekomponenten.  |

Systemqualitäten sind eng miteinander verknüpft. Die Anforderungen für eine Systemqualität können die Anforderungen und das Konzept für andere Systemqualitäten beeinflussen. So kann beispielsweise ein höheres Sicherheitsniveau die Leistung beeinträchtigen, was wiederum die Verfügbarkeit beeinflusst. Das Hinzufügen zusätzlicher Server, um Verfügbarkeitsprobleme zu beheben, wirkt sich auf die Zweckmäßigkeit (Wartungskosten) aus.

Das Verständnis, wie die einzelnen Systemqualitäten miteinander in Beziehung stehen und wie diese abgestimmt werden müssen, ist der Schlüssel für die Entwicklung eines Systems, das sowohl den Geschäftsanforderungen als auch den Geschäftseinschränkungen erfolgreich gerecht wird.

In den folgenden Abschnitten werden die Systemqualitäten, die sich auf das Bereitstellungskonzept auswirken, näher beschrieben. Es wird außerdem erläutert, welche Faktoren bei der Gestaltung der QoS-Anforderungen berücksichtigt werden müssen. Ein Abschnitt zu den Anforderungen an die Dienstebenen, die die Grundlage für die Vereinbarungen auf Dienstebene (Service Level Agreement, SLA) bilden, ist ebenfalls enthalten.

## Leistung

Geschäftsanforderungen messen die Leistung normalerweise in nicht technischen Begriffen, die sich auf die Antwortzeit beziehen. Eine Geschäftsanforderung für den internetbasierten Zugriff könnte beispielsweise wie folgt lauten:

Die Benutzer erwarten eine angemessene Antwortzeit bei der Anmeldung von üblicherweise maximal vier Sekunden.

Untersuchen Sie, beginnend mit dieser Geschäftsanforderung, sämtliche Anwendungsfälle, um zu bestimmen, wie diese Anforderung auf einer Systemebene ausgedrückt werden könnte. In einigen Fällen möchten Sie möglicherweise die



Bedingungen der Benutzerauslastung miteinschließen, die während der Anwendungsanalyse ermittelt wurden. Formulieren Sie die Leistungsanforderung für jeden Anwendungsfall in Bezug auf die Antwortzeit unter bestimmten Auslastungsbedingungen oder in Bezug auf die Antwortzeit zuzüglich Datendurchsatz. Sie können auch die zulässige Anzahl an Fehlern angeben.

Im Folgenden wird anhand von zwei Beispielen verdeutlicht, wie die Systemanforderungen für die Leistung angegeben werden können:

- Die Antwortzeit bei der Webseitenaktualisierung darf während des gesamten Tages 4 Sekunden nicht überschreiten (Prüfung alle 15 Minuten), wobei nur weniger als 3,4 Fehler pro eine Million Transaktionen auftreten dürfen.
- Während definierter Spitzenzeiträume muss das System 25 sichere Anmeldungen pro Sekunde zulassen, wobei die Antwortzeit für jeden Benutzer 12 Sekunden nicht überschreiten darf und weniger als 3,4 Fehler pro eine Million Transaktionen auftreten dürfen.

Leistungsanforderungen sind eng mit Verfügbarkeitsanforderungen (Failover-Auswirkung auf die Leistung) und der latenten Kapazität (Menge der verfügbaren Kapazität für die Bewältigung außergewöhnlicher Spitzenauslastung) verknüpft.

## Verfügbarkeit

Über die Verfügbarkeit kann man die Betriebszeit eines Systems bestimmt werden. Sie bemisst sich normalerweise aus dem Prozentsatz der Zeit, während der Benutzer auf das System zugreifen können. Die Zeit, in der das System nicht verfügbar ist (Ausfallzeit) kann durch Ausfälle der Hardware, der Software, des Netzwerks oder durch einen beliebigen anderen Faktor (z. B. Unterbrechung der Stromversorgung) verursacht werden, durch den das System zum Stillstand kommt. Geplante Zeiten, in denen das System zu Servicezwecken (Wartung und Aufrüstung) heruntergefahren wird, werden nicht als Ausfallzeit bezeichnet. Eine grundlegende Gleichung zur Berechnung der Systemverfügbarkeit als Prozentsatz der Betriebszeit lautet wie folgt:

$$\text{Verfügbarkeit} = \text{Uptime} / (\text{Uptime} + \text{Downtime}) * 100\%$$

Üblicherweise messen Sie die Verfügbarkeit anhand der Anzahl der „Neunen“, die Sie erreichen können. Bei einer 99%-igen Verfügbarkeit erhalten Sie zwei Neunen. Die Angabe zusätzlicher Neunen wirkt sich erheblich auf das Bereitstellungs-konzept aus. In der folgenden Tabelle wird die unvorhergesehene Ausfallszeit bei zusätzlichen „Verfügbarkeits-Neunen“ auf einem System berechnet, das 24 Stunden an 7 Tagen die Woche ganzjährig (insgesamt 8.760 Stunden) verfügbar ist.

**TABELLE 3-3** Unvorhergesehene Ausfallzeit für ein System, das ganzjährig (8.760 Stunden) verfügbar ist

| Anzahl der Neunen | Verfügbarkeit (Prozent) | Unvorhergesehene Ausfallzeit |
|-------------------|-------------------------|------------------------------|
| 2                 | 99 %                    | 88 Stunden                   |
| 3                 | 99,9 %                  | 9 Stunden                    |
| 4                 | 99,99 %                 | 45 Minuten                   |
| 5                 | 99,999 %                | 5 Minuten                    |

## Fehlertolerante Systeme

Verfügbarkeitsanforderungen von vier oder fünf Neunen erfordern im Normalfall ein fehlertolerantes System. Ein fehlertolerantes System muss in der Lage sein, seinen Dienst auch während eines Hardware- oder Softwareausfalls fortzusetzen. Normalerweise wird die Fehlertoleranz durch Redundanz in der Hardware (CPUs, Speicher- und Netzwerkgeräte) als auch in der Software, die die wichtigsten Dienste zur Verfügung stellt, erreicht.

Bei einem Single-Point-of-Failure handelt es sich um eine Hardware- oder Softwarekomponente, die Teil eines wichtigen Pfads ist, aber nicht von redundanten Komponenten gesichert wird. Der Ausfall dieser Komponente führt zum Dienstausfall des Systems. Wenn Sie ein fehlertolerantes System konzipieren, müssen Sie potenzielle Single-Point-of-Failure ermitteln und vermeiden.

Fehlertolerante Systeme können in der Implementierung und Wartung teuer sein. Stellen Sie sicher, dass Sie verstanden haben, worin die Geschäftsanforderungen für die Verfügbarkeit liegen, und berücksichtigen Sie die Strategien und Kosten der Verfügbarkeitslösungen, die diese Anforderungen erfüllen.

## Priorisieren der Dienstverfügbarkeit

Aus Sicht des Benutzers bezieht sich die Verfügbarkeit oft eher auf einzelne Dienste als auf die Verfügbarkeit des gesamten Systems. So hat es beispielsweise im Normalfall keine oder nur geringe Auswirkungen auf andere Dienste, wenn keine Instant Messaging-Dienste verfügbar sind. Wenn jedoch Dienste nicht verfügbar sind, von denen viele andere Dienste abhängig sind (z. B. Directory Server), sind die Auswirkungen weitreichender. Genauere Verfügbarkeitspezifikationen sollten deutlich auf bestimmte Anwendungsfälle und Anwendungsanalysen verweisen, für die eine höhere Verfügbarkeit erforderlich ist.

Es ist hilfreich, Verfügbarkeitsbedürfnisse entsprechend den verschiedenen aufgestellten Prioritäten aufzulisten. In der folgenden Tabelle wird die Verfügbarkeit verschiedener Dienstypen nach Priorität geordnet aufgeführt.

**TABELLE 3–4** Verfügbarkeit von Diensten nach Priorität

| Priorität | Diensttyp              | Beschreibung  |
|-----------|------------------------|---|
| 1         | Unerlässlich           | Dienste, die immer verfügbar sein müssen. Beispielsweise Datenbankdienste (z. B. LDAP-Verzeichnisse) für Anwendungen.   |
| 2         | Muss verfügbar sein    | Dienste, die verfügbar sein müssen, aber mit reduzierter Leistung verfügbar sein können. Die Verfügbarkeit von Messaging-Diensten ist möglicherweise in einigen Unternehmensumgebungen nicht zwingend erforderlich. |
| 3         | Kann verschoben werden | Dienste, die innerhalb eines angegebenen Zeitraums verfügbar sein müssen. Die Verfügbarkeit von Kalenderdiensten ist möglicherweise in einigen Unternehmensumgebungen nicht erforderlich.                           |
| 4         | Optional               | Dienste, die auf unbegrenzte Zeit verschoben werden können. In einigen Umgebungen können Instant Messaging-Dienste zwar als nützlich aber nicht unbedingt als erforderlich bewertet werden.                         |

## Dienstausfall

Im Rahmen des Verfügbarkeitskonzepts werden auch die Auswirkungen berücksichtigt, die entstehen, wenn die Verfügbarkeit beeinträchtigt wird oder eine Komponente ausfällt. Hierbei muss auch berücksichtigt werden, ob die angemeldeten Benutzer Sitzungen neu starten müssen und inwiefern ein Fehler in einem Bereich sich auf andere Bereiche eines Systems auswirkt. QoS-Anforderungen sollten diese Szenarien beachten und angeben, wie die Bereitstellung auf derartige Situationen reagiert.

## Skalierbarkeit

Unter Skalierbarkeit wird die Fähigkeit verstanden, einem System Kapazitäten hinzuzufügen, sodass das System eine zusätzliche Auslastung durch bestehende Benutzer oder durch einen erweiterten Benutzerstamm unterstützen kann. Die Skalierbarkeit setzt normalerweise eine Erweiterung der Ressourcen voraus; es sollten jedoch keine Änderungen am Konzept der Bereitstellungsarchitektur oder Dienstausfälle erforderlich sein, die sich aufgrund der für das Hinzufügen zusätzlicher Ressourcen benötigten Zeit ergeben.

Wie auch die Verfügbarkeit bezieht sich die Skalierbarkeit eher auf einzelne von einem System bereitgestellte Dienste als auf das gesamte System. Im Fall von Diensten, von denen andere Dienste abhängig sind (z. B. Directory Server), kann sich die Skalierbarkeit auf das gesamte System auswirken.

Die Skalierbarkeitsanforderungen werden nicht notwendigerweise mit den QoS-Anforderungen angegeben, es sei denn, das geplante Wachstum der Bereitstellung wird eindeutig in den Geschäftsanforderungen festgehalten. Während der Bereitstellungs-konzeptphase des Lösungslebenszyklus sollte die Bereitstellungsarchitektur grundsätzlich einen gewissen Spielraum für die Skalierung des Systems einschließen, auch wenn keine QoS-Anforderungen für die Skalierbarkeit angegeben wurden.

## Wachstumseinschätzung

Bei der Einschätzung des Wachstums eines Systems zur Ermittlung der Skalierbarkeitsanforderungen muss mit Hochrechnungen, Schätzungen und Vermutungen gearbeitet werden, die möglicherweise nicht eintreten. Es gibt folgende drei Schlüssel für die Entwicklung von Anforderungen für ein skalierbares System:

- **Hochleistungskonzept-Strategie.** Bei der Spezifikation der Leistungsanforderungen sollte die latente Kapazität einbezogen werden, um Auslastungen zu berücksichtigen, die im Laufe der Zeit zunehmen. Maximieren Sie außerdem die Verfügbarkeit innerhalb der Budgeteinschränkungen. Durch diese Strategie können Sie das Wachstum absorbieren und Meilensteine für die Skalierung des Systems besser setzen.
- **Inkrementelle Bereitstellung.** Durch eine inkrementelle Bereitstellung können Sie besser planen, wann Ressourcen hinzugefügt werden. Geben Sie eindeutige Meilensteine für die Skalierung des Systems an. Meilensteine sind normalerweise lastenbasierte Anforderungen, die mit bestimmten Daten für die Prüfung der Skalierbarkeit koordiniert werden.
- **Umfassende Leistungsüberwachung.** Durch die Überwachung der Leistung können Sie besser festlegen, wann dem System Ressourcen hinzugefügt werden sollten. Anforderungen für die Überwachung der Leistung können einen Leitfaden für Operatoren und Administratoren darstellen, die für die Wartung und Aktualisierung verantwortlich sind.

In der folgenden Tabelle werden die Faktoren aufgeführt, die bei der Ermittlung der Skalierbarkeitsanforderungen berücksichtigt werden sollten.

**TABELLE 3–5** Skalierbarkeitsfaktoren

| Thema                        | Beschreibung  |
|------------------------------|---|
| Analyse der Anwendungsmuster | Machen Sie sich mithilfe der vorhandenen Daten mit den Anwendungsmustern des aktuellen (oder geplanten) Benutzerstamms vertraut. Falls keine aktuellen Daten vorliegen, analysieren Sie die Branchendaten oder Markteinschätzungen. |

**TABELLE 3-5** Skalierbarkeitsfaktoren (Fortsetzung)

| Thema  | Beschreibung  |
|--|---|
| Konzept in Bezug auf maximal sinnvollen Umfang | <p>Konzepterstellung mit Hinblick auf den maximal erforderlichen Umfang sowohl für den bekannten als auch für den möglichen Bedarf.</p> <p>Hierbei handelt es sich häufig um eine Schätzung über 24 Monate, der die Leistungseinschätzung der vorhandenen Benutzerauslastung und angemessene Eischätzungen für die zukünftige Auslastung zugrunde liegen. Der Zeitraum für die Schätzung hängt in hohem Maße von der Zuverlässigkeit der Prognosen ab.</p>  |
| Festlegen angemessener Meilensteine            | <p>Schrittweise Implementierung des Bereitstellungskonzepts, um kurzfristige Anforderungen zu erfüllen, wobei ein Puffer für unerwartete Wachstumsschübe eingebaut wird. Legen Sie Meilensteine für das Hinzufügen von Systemressourcen fest.</p> <p>Beispiel:</p> <ul style="list-style-type: none"><li>■ Kapitalbeschaffung (z. B. vierteljährlich oder jährlich)</li><li>■ Vorlaufzeit für den Erwerb von Hardware oder Software (z. B. eine bis sechs Wochen)</li><li>■ Puffer (10 % bis 100 %, je nach Wachstumserwartungen)</li></ul> |
| Aufnahme neuer Technologien                    | <p>Machen Sie sich mit neuen Technologien, wie beispielsweise schnelleren Prozessoren und Webservern, vertraut und darüber, wie die jeweilige Technologie sich auf die Leistung der zugrunde liegenden Architektur auswirkt.</p>  |

## Sicherheitsanforderungen

Sicherheit ist ein komplexes Thema, das alle Stufen eines bereitgestellten Systems betrifft. Die Entwicklung von Sicherheitsanforderungen schließt die Identifizierung der Sicherheitsbedrohungen und die Entwicklung einer Strategie zur Bekämpfung dieser Gefahren ein. Diese Sicherheitsanalyse beinhaltet folgende Schritte:

1. Identifizierung unerlässlicher Aspekte
2. Identifizierung der Bedrohungen dieser Aspekte
3. Identifizierung von Anfälligkeiten, aufgrund derer die Bedrohungen das Unternehmen gefährden können
4. Entwicklung eines Sicherheitsplans, mit dessen Hilfe die Gefahr für das Unternehmen eingeschränkt wird

Bei der Analyse der Sicherheitsanforderungen sollte ein Querschnitt durch die Interessengruppen in Ihrem Unternehmen, wie beispielsweise leitende Angestellte, Unternehmensanalysten und Mitarbeiter aus dem IT-Bereich, einbezogen werden. Häufig wird in einem Unternehmen ein Sicherheitsarchitekt ernannt, der für die Entwicklung und Implementierung der Sicherheitsmaßnahmen verantwortlich ist.

Im folgenden Abschnitt werden einige der Bereiche beschrieben, die bei der Sicherheitsplanung abgedeckt werden.

## Elemente eines Sicherheitsplans

Die Planung der Sicherheit eines Systems ist Bestandteil des Bereitstellungskonzepts, das für die erfolgreiche Implementierung unabdingbar ist. Ziehen Sie bei der Sicherheitsplanung folgende Aspekte in Betracht:

- **Physische Sicherheit.** Die physische Sicherheit betrifft den physischen Zugriff auf Router, Server, Serverräume, Datenzentren und andere Teile der Infrastruktur. Andere Sicherheitsmaßnahmen werden beeinträchtigt, wenn eine nicht autorisierte Person einen Serverraum betreten und dort die Routerverbindungen trennen kann.
- **Netzwerksicherheit.** Die Netzwerksicherheit betrifft den Zugriff auf Ihr Netzwerk über Firewalls, sichere Zugriffsbereiche, Zugriffssteuerungslisten und den Portzugriff. Um die Netzwerksicherheit herzustellen, entwickeln Sie Strategien gegen unberechtigte Zugriffe, Sabotage und Dienstverweigerungs-(DoS-)Angriffe.
- **Anwendungs- und Anwendungsdatensicherheit.** Die Anwendungs- und Anwendungsdatensicherheit deckt den Zugriff auf Benutzerkonten, Unternehmensdaten und Unternehmensanwendungen mithilfe von Authentifizierungs- und Berechtigungsverfahren und -richtlinien ab. Dieser Bereich beinhaltet die Definition folgender Richtlinien:
  - Passwortrichtlinien
  - Zugriffsrechte, wie beispielsweise delegierte Administratorrechte für Benutzer im Gegensatz zum Administratorzugriff
  - Kontodeaktivierung
  - Zugriffssteuerung
  - Verschlüsselungsrichtlinien, wie beispielsweise sichere Datenübertragung und Verwendung von Zertifikaten zur Datensignierung
- **Persönliche Sicherheitspraktiken.** Eine unternehmensübergreifende Sicherheitsrichtlinie definiert die Arbeitsumgebung und die Methoden, denen alle Benutzer unterworfen sind, um sicherzustellen, dass andere Sicherheitsmaßnahmen wie geplant funktionieren. Im Normalfall entwickeln Sie einen Sicherheitsleitfaden oder ein Handbuch und bieten Schulungen zu den Sicherheitspraktiken für die Benutzer an. Um eine effektive umfassende Sicherheitsstrategie zu gewährleisten, müssen vernünftige Sicherheitsmaßnahmen Teil der Unternehmenskultur werden.

## Latente Kapazität

Die latente Kapazität ist die Fähigkeit einer Bereitstellung, eine außergewöhnliche Spitzenauslastung zu bewältigen, ohne zusätzliche Ressourcen hinzuzufügen. Normalerweise geben Sie QoS-Anforderungen nicht direkt über die latente Kapazität an, aber diese Systemqualität ist ein Faktor für die Verfügbarkeit, Leistung und Skalierbarkeit des Systems.

# Wartungseignungsanforderungen

Die Wartungseignung ist die Einfachheit, mit der ein bereitgestelltes System gewartet werden kann. Dazu zählen Aufgaben wie Überwachung des Systems, Behebung von auftretenden Problemen, Hinzufügen zum und Entfernen von Benutzern vom System und Aufrüsten der Hardware- und Softwarekomponenten.

Beim Planen der Wartungseignungsanforderungen sollten Sie die in der folgenden Tabelle aufgeführten Aspekte berücksichtigen.

**TABELLE 3-6** Aspekte für die Wartungseignungsanforderungen

| Thema                     | Beschreibung   |
|---------------------------|--|
| Planung von Ausfallzeiten | <p>Identifizierung der Wartungsaufgaben, für die bestimmte Dienste nicht verfügbar oder teilweise nicht verfügbar sein dürfen.</p> <p>Einige Wartungs- und Aufrüstungsaufgaben können nahtlos für den Benutzer erfolgen, während für andere der Dienst unterbrochen werden muss. Planen Sie, wenn möglich, die Wartungsaufgaben, die eine Ausfallzeit mit sich bringen, zusammen mit den Benutzern, sodass die Benutzer sich auf die Ausfallzeiten einstellen können.</p>  |
| Anwendungsmuster          | <p>Identifizierung der Anwendungsmuster, um den besten Zeitpunkt für die Wartung zu bestimmen.</p> <p>Beispielsweise sollten Sie die Wartung von Systemen, deren Spitzenauslastung zu den normalen Geschäftszeiten ist, für die Abende oder Wochenenden planen. Bei geografisch verteilten Systemen kann die Ermittlung dieser Zeitpunkte schwieriger sein.</p>  |
| Verfügbarkeit             | <p>Die Wartungseignung reflektiert oft Ihr Verfügbarkeitskonzept. Die Strategien zur Minimierung der Ausfallzeiten für Wartungs- und Aufrüstungsaufgaben hängen von der Verfügbarkeitsstrategie ab. Bei Systemen, die häufig verfügbar sein müssen, gibt es weniger Gelegenheiten für Wartung, Aufrüstung und Reparaturen.</p> <p>Strategien für die Handhabung der Verfügbarkeitsanforderungen bestimmen, wie Wartungs- und Aufrüstungsaufgaben gehandhabt werden. Bei Systemen, die geografisch verteilt sind, hängt die Serviceplanung etwa davon ab, ob die Arbeitslast während der Wartungszeiträume an remote Server weitergeleitet werden kann.</p> <p>Außerdem sind für Systeme, die in hohem Maße verfügbar sein müssen, kompliziertere Lösungen erforderlich, bei denen die Systeme automatisch neu gestartet werden und der Benutzer nur geringfügig eingreifen muss.</p> |

**TABELLE 3-6** Aspekte für die Wartungseignungsanforderungen (Fortsetzung)

| Thema                    | Beschreibung   |
|--------------------------|--|
| Diagnose und Überwachung | <p>Sie können die Stabilität eines Systems verbessern, indem Sie regelmäßig Diagnose- und Überwachungstools ausführen, um Problembereiche zu identifizieren.</p> <p>Die regelmäßige Überwachung eines Systems kann Probleme noch vor ihrem Auftreten vermeiden, die Arbeitsauslastung an den Verfügbarkeitsstrategien ausrichten und die Planung für Wartung und Ausfallzeiten verbessern.</p> |

---

## Anforderungen an die Dienstebene

In einer Vereinbarung auf Dienstebene (Service Level Agreement, SLA) werden die leistungsbezogenen Mindestanforderungen angegeben. Wenn diese Anforderungen nicht erfüllt werden, geht hieraus ebenfalls hervor, auf welcher Ebene und in welchem Umfang Kundenunterstützung bereitgestellt werden muss. Anforderungen an die Dienstebene sind Systemanforderungen, die die Bedingungen angeben, auf denen die SLA basiert.

Wie auch QoS-Anforderungen werden Anforderungen an die Dienstebene von den Geschäftsanforderungen abgeleitet und garantieren die allgemeine Systemqualität, die das bereitgestellte System aufweisen muss. Da die Vereinbarung auf Dienstebene als Vertrag angesehen wird, sollten die Anforderungen an die Dienstebene eindeutig festgelegt sein. Die Anforderungen an die Dienstebene definieren genau, unter welchen Bedingungen die Anforderungen getestet werden und welche Folgen es nach sich zieht, wenn die Anforderungen nicht erfüllt werden.



## Logisches Konzept

---

Während der Phase des logischen Konzepts im Lebenszyklus einer Lösung entwerfen Sie eine logische Architektur, in der die gegenseitigen Abhängigkeiten der logischen Komponenten der Lösung dargestellt werden. Die logische Architektur und die Anwendungsanalyse aus der Phase der technischen Anforderungen bilden ein Bereitstellungsszenario, das als Ausgangspunkt für die Konzeptphase der Bereitstellung dient.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Informationen zu logischen Architekturen“ auf Seite 49
- „Entwurf einer logischen Architektur“ auf Seite 51
- „Java Enterprise System-Komponenten“ auf Seite 51
- „Beispiel für logische Architekturen“ auf Seite 58
- „Zugriffszonen“ auf Seite 67
- „Bereitstellungsszenario“ auf Seite 68

---

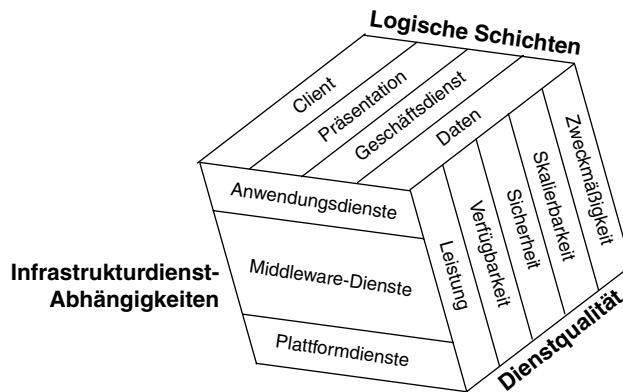
## Informationen zu logischen Architekturen

In einer logischen Architektur werden die Softwarekomponenten identifiziert, die für die Implementierung einer Lösung erforderlich sind, und in ihrer gegenseitigen Abhängigkeit dargestellt. Die logische Architektur und die Dienstqualitätsanforderungen, die während der Phase der technischen Anforderungen bestimmt wurden, bilden ein Bereitstellungsszenario. Das Bereitstellungsszenario bildet die Grundlage für den Entwurf der Bereitstellungsarchitektur, der in der nächsten Phase im Rahmen des Bereitstellungskonzepts stattfindet.

Bei der Entwicklung einer logischen Architektur müssen nicht nur die Komponenten identifiziert werden, die Dienste für die Benutzer bereitstellen, sondern auch Komponenten, die die erforderlichen Middleware- und Plattformdienste zur Verfügung stellen. Infrastrukturdienst-Abhängigkeiten und logische Schichten bieten zwei einander ergänzende Möglichkeiten zur Durchführung dieser Analyse.

Infrastrukturdienst-Abhängigkeiten und logische Schichten sind zwei von drei Pfeilern der Lösungsarchitektur, auf denen Sun Java™ Enterprise System basiert. Diese drei Pfeiler werden nachfolgend aufgeführt und werden zudem in „Informationen zu logischen Architekturen“ auf Seite 49 dargestellt.

- **Infrastrukturdienst-Abhängigkeiten.** Interagierende Softwarekomponenten, mit denen Unternehmensdienste bereitgestellt werden. Für die Softwarekomponenten ist eine Reihe von zugrunde liegenden Infrastrukturdiensten erforderlich, über die die verteilten Komponenten miteinander kommunizieren und interagieren können.
- **Logische Schichten.** Eine logische Anordnung von Softwarekomponenten in Schichten, die die logische und physische gegenseitige Abhängigkeit der Softwarekomponenten auf der Grundlage der von ihnen bereitgestellten Dienste darstellen.
- **Dienstqualität.** Dienstqualitäten des Systems, wie Leistung, Verfügbarkeit, Skalierbarkeit und sonstige Eigenschaften, die bestimmte Aspekte des Konzepts und der Funktion einer Softwarelösung darstellen.




---

**Hinweis** – Weitere Informationen zu den Java Enterprise System-Architekturkonzepten finden Sie in Kapitel „Java Enterprise System Architecture“ im *Sun Java Enterprise System 2005Q4 Technischer Überblick*.

---

In einer logischen Architektur werden Dienstebenen dargestellt, indem die erforderlichen Komponenten und deren jeweilige Abhängigkeiten abgebildet werden. In einer logischen Architektur werden zudem die Komponenten in logische Schichten

aufgeteilt, in denen Präsentations-, Geschäfts- und Datendienste dargestellt werden, auf die über eine Client-Schicht zugegriffen werden kann. Die Dienstqualitätsanforderungen werden in der logischen Architektur nicht entworfen, sie werden jedoch im Bereitstellungsszenario mit der logischen Architektur kombiniert.

---

## Entwurf einer logischen Architektur

Verwenden Sie zum Entwerfen einer logischen Architektur die Anwendungsfälle, die während der Phase der technischen Anforderungen bestimmt wurden, um die Java Enterprise System-Komponenten zu identifizieren, die die für die Lösung erforderlichen Dienste bereitstellen. Zudem müssen Sie alle Komponenten bestimmen, die Dienste für die ursprünglich identifizierten Komponenten bereitstellen.

Sie platzieren die Java Enterprise System-Komponenten entsprechend der von Ihnen bereitgestellten Art von Diensten innerhalb des Kontexts einer mehrschichtigen Architektur. Das Verständnis der Komponenten als Bestandteil einer mehrschichtigen Architektur hilft Ihnen später dabei, die Verteilung der von den Komponenten bereitgestellten Dienste zu bestimmen und eine Strategie zur Implementierung der Dienstqualität festzulegen (wie z. B. Skalierbarkeit, Verfügbarkeit usw.).

Zusätzlich können Sie eine andere Ansicht der logischen Komponenten angeben, in der diese innerhalb sicherer Zugriffszonen platziert werden. Im Abschnitt „Zugriffszonen“ auf Seite 67 finden Sie ein Beispiel für sichere Zugriffszonen.

---

## Java Enterprise System-Komponenten

Java Enterprise System besteht aus interagierenden Softwarekomponenten, die Dienste bereitstellen, die Sie zur Erstellung Ihrer Unternehmenslösung verwenden können. In der folgenden Abbildung werden die wichtigsten in Java Enterprise System enthaltenen Softwarekomponenten dargestellt. Im Handbuch *Sun Java Enterprise System 2005Q4 Technischer Überblick* finden Sie zusätzliche Informationen zu Java Enterprise System-Komponenten und zu den durch die Komponenten bereitgestellten Diensten.

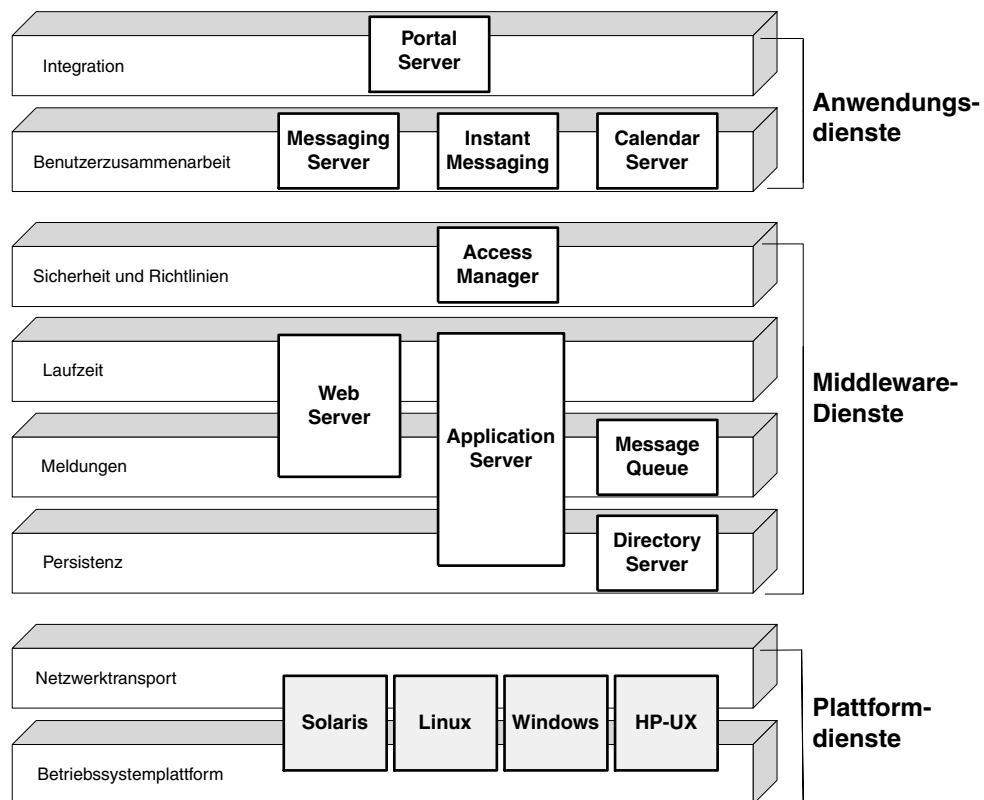


ABBILDUNG 4-1 Java Enterprise System-Komponenten

## Komponentenabhängigkeiten

Bei der Identifizierung der Java Enterprise System-Komponenten für eine logische Architektur müssen Sie auch die Komponenten für die Unterstützung ermitteln. Wenn Sie beispielsweise Messaging Server als eine erforderliche Komponente in einer logischen Architektur ermitteln, muss die logische Architektur ebenfalls Directory Server und möglicherweise Access Manager enthalten. Messaging Server ist hinsichtlich der Verzeichnisdienste von Directory Server und für Lösungen, die Single Sign-On erfordern, von Access Manager abhängig.

In der folgenden Tabelle werden die Abhängigkeiten von Java Enterprise System-Komponenten aufgelistet. Eine grafische Darstellung der Abhängigkeiten zwischen den Schlüsselkomponenten finden Sie unter „[Komponentenabhängigkeiten](#)“ auf Seite 52. Verwenden Sie beim Entwurf einer logischen Architektur folgende Tabelle und die dazugehörige Abbildung, um abhängige Komponenten in Ihrem Konzept zu bestimmen.

**TABELLE 4–1** Java Enterprise System-Komponentenabhängigkeiten

| Java Enterprise System-Komponente  | Ist abhängig von  |
|------------------------------------|---|
| Application Server                 | Message QueueDirectory Server (optional)  |
| Calendar Server                    | Messaging Server (für den E-Mail-Benachrichtigungsdienst) Access Manager (für Single Sign-on) Web Server (für die Webschnittstelle) Directory Server  |
| Communications Express             | Access Manager (für Single Sign-on) Calendar ServerMessaging ServerInstant MessagingWeb Server (für die Webschnittstelle) Directory Server  |
| Directory Proxy Server             | Directory Server  |
| Directory Server                   | Keine   |
| Access Manager                     | Application Server oder Web ServerDirectory Server  |
| Instant Messaging                  | Access Manager (für Single Sign-On) Directory Server  |
| Message Queue                      | Directory Server (optional)   |
| Messaging Server                   | Access Manager (für Single Sign-On) Web Server (für die Webschnittstelle) Directory Server  |
| Portal Server                      | Wenn zur Verwendung von Portal Server-Kanälen konfiguriert:<br><br>Calendar ServerMessaging ServerInstant Messaging<br><br>Access Manager (für Single Sign-On) Application Server oder Web ServerDirectory Server |
| Portal Server Secure Remote Access | Portal Server   |
| Web Server                         | Access Manager (optional, für die Zugriffssteuerung)  |

**Hinweis** – Die Abhängigkeiten zwischen den in der Tabelle „Komponentenabhängigkeiten“ auf Seite 52 aufgelisteten Java Enterprise System-Komponenten sind nicht vollständig. In der Tabelle „Komponentenabhängigkeiten“ auf Seite 52 werden keine Abhängigkeiten aufgelistet, die bei der Planung der Installation zu berücksichtigen sind. Eine vollständige Liste der Java Enterprise System-Abhängigkeiten finden Sie im *Sun Java Enterprise System 2005Q4 Installationshandbuch für UNIX*.

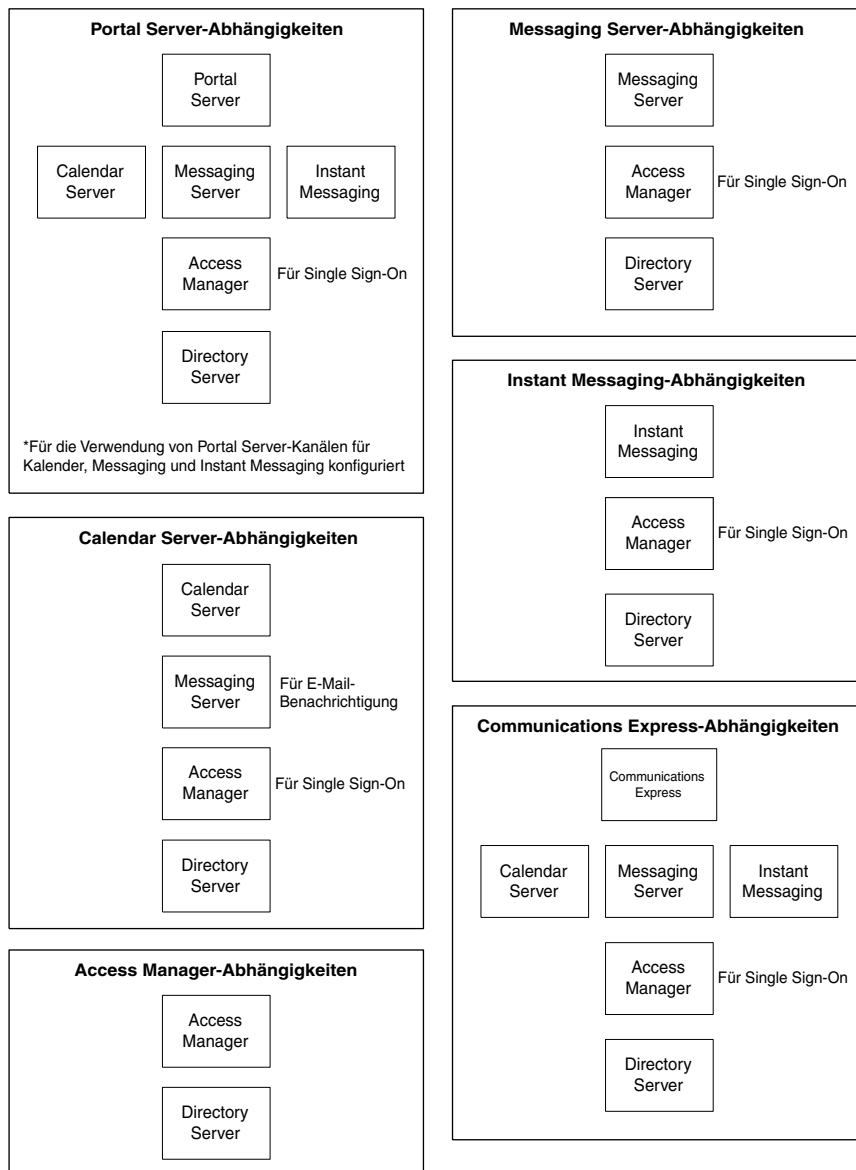


ABBILDUNG 4-2 Java Enterprise System-Komponentenabhängigkeiten

## Webcontainer-Unterstützung

Im vorigen Abschnitt „Komponentenabhängigkeiten“ auf Seite 52 wurde der Webcontainer, in dem Portal Server und Access Manager ausgeführt werden, nicht berücksichtigt. Dieser Webcontainer kann durch Application Server, Web Server oder ein Drittanbieterprodukt bereitgestellt werden. Beim Entwurf einer logischen Architektur, die Portal Server oder Access Manager enthält, sollten Sie sicherstellen, dass der für diese Komponenten erforderliche Webcontainer berücksichtigt wird.

## Von Messaging Server bereitgestellte logisch eindeutige Dienste

Der Java Enterprise System Messaging Server kann für die Bereitstellung der folgenden logischen eindeutigen Dienste konfiguriert werden:

- Message Transfer Agent
- Message Multiplexor
- Message Express Multiplexor
- Nachrichtenspeicher

Diese unterschiedliche Konfigurationen von Messaging Server bieten Funktionen, die auf separaten physischen Servern bereitgestellt und in verschiedenen Schichten einer logischen Architektur dargestellt werden können. Da diese Konfigurationen für Messaging Server logisch eindeutige Dienste in separaten Schichten darstellen, können Sie diese beim Entwurf einer logischen Architektur als logisch eindeutige Komponenten betrachten. Im Abschnitt „Beispiel für logische Architekturen“ auf Seite 58 finden Sie ein Beispiel für logisch eindeutige Komponenten.

In der folgenden Tabelle werden die logisch eindeutigen Konfigurationen von Messaging Server beschrieben.

**TABELLE 4-2** Messaging Server-Konfigurationen

| Unterkomponente              | Beschreibung  |
|------------------------------|---|
| Message Transfer Agent (MTA) | Unterstützt das Senden von E-Mails durch das Bearbeiten von SMTP-Verbindungen, die Weiterleitung von E-Mails und die Zustellung von Nachrichten an die richtigen Nachrichtenspeicher. Die MTA-Komponenten können für die Unterstützung von E-Mails konfiguriert werden, die von Standorten außerhalb des Unternehmens (eingehend) oder die von Standorten innerhalb des Unternehmens (ausgehend) gesendet werden. |
| Nachrichtenspeicher (STR)    | Ermöglicht das Abrufen und Speichern von E-Mail-Nachrichten.  |

**TABELLE 4–2** Messaging Server-Konfigurationen (Fortsetzung)

| Unterkomponente                     | Beschreibung   |
|-------------------------------------|--|
| Message Multiplexor (MMP)           | Unterstützt das Abrufen von E-Mails durch Zugriff auf den Nachrichtenspeicher für E-Mail-Clients unter Verwendung des IMAP- oder POP-Protokolls. |
| Messenger Express Multiplexor (MEM) | Unterstützt den Abruf von E-Mails durch Zugriff auf den Nachrichtenspeicher im Namen webbasierter (HTTP)-Clients.                                |

## Zugriffskomponenten

Java Enterprise System enthält zudem Komponenten, die häufig von außerhalb der Firewall eines Unternehmens Zugriff auf Systemdienste bieten. Mit manchen Konfigurationen von Messaging Server kann darüber hinaus Netzwerkzugriff gewährt werden, beispielsweise wenn Messaging Server für Message Multiplexor konfiguriert wird. In der folgenden Tabelle werden die Java Enterprise System-Komponenten aufgeführt, die Remote-Zugriff auf Systemdienste gewähren.

**TABELLE 4–3** Java Enterprise System-Komponenten, die Remote-Zugriff bieten

| Komponente  | Beschreibung  |
|---|---|
| Directory Proxy Server                            | Bietet eine verbesserte Steuerung des Verzeichniszugriffs, der Schemakompatibilität, der Weiterleitung und Lastenverteilung für mehrere Directory Server-Instanzen.                     |
| Portal Server, Portal Server Secure Remote Access | Ermöglicht außerhalb einer Unternehmens-Firewall den sicheren Internetzugriff auf den Inhalt und die Dienste von Portal Server, einschließlich interne Portale und Internetanwendungen. |
| Portal Server, Portal Server Mobile Access        | Bietet drahtlosen Zugriff über mobile Geräte und Sprachzugriff auf Portal Server.   |
| Messaging Server Message Multiplexor (MMP)        | Unterstützt den Abruf von E-Mails durch Zugriff auf den Nachrichtenspeicher im Namen webbasierter (HTTP)-Clients.   |

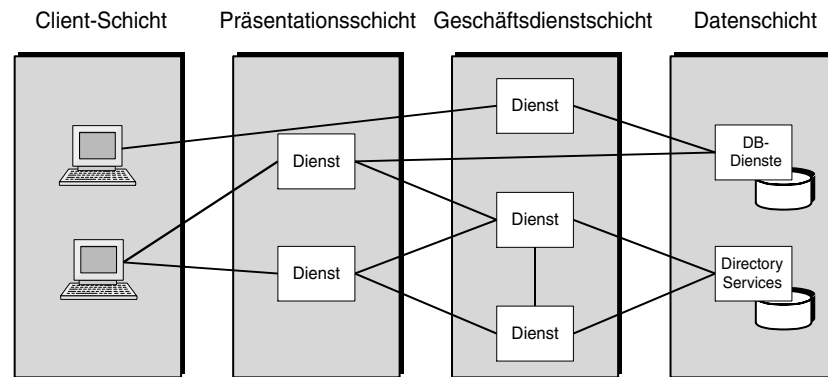
Komponenten, die Remote-Zugriff ermöglichen, werden in der Regel in sicheren Zugriffszonen bereitgestellt, wie im Beispiel in Abschnitt „Zugriffszonen“ auf Seite 67 angegeben.

## Entwurf einer mehrschichtigen Architektur

Java Enterprise System eignet sich für den Entwurf einer mehrschichtigen Architektur, in der Dienste entsprechend den von ihnen bereitgestellten Funktionen in Schichten platziert werden. Alle Dienste sind logisch unabhängig und können von Diensten aufgerufen werden, die sich entweder in derselben oder in einer anderen Schicht



befinden. In der folgenden Abbildung wird ein mehrschichtiges Architekturmodell für Unternehmensanwendungen dargestellt, in dem der Client, die Präsentation, der Geschäftsdienst und die Datenschichten enthalten sind.



**ABBILDUNG 4-3** Mehrschichtiges Architekturmodell

In der folgenden Tabelle werden die in „[Entwurf einer mehrschichtigen Architektur](#)“ auf Seite 56 dargestellten logischen Schichten beschrieben.

**TABELLE 4-4** Logische Schichten in einer mehrschichtigen Architektur

| Schicht                 | Beschreibung   |
|-------------------------|--|
| Client-Schicht          | Enthält die Client-Anwendungen, die den Endbenutzern Informationen anzeigen. Für Java Enterprise System sind diese Anwendungen in der Regel Mail-Clients, Webbrowser oder Mobile Access-Clients.   |
| Präsentationsschicht    | Stellt Dienste bereit, die Endbenutzern Daten anzeigen und ihnen die Verarbeitung und Änderung der Präsentation ermöglichen. Mit einem Webmail-Client oder einer Portal Server-Komponente können die Benutzer beispielsweise die Darstellung der empfangenen Informationen ändern.   |
| Geschäftsdienstschiicht | Bietet Back-End-Dienste, mit denen in der Regel Daten aus der Datenschicht abgerufen werden, um die Daten für andere Dienste innerhalb der Präsentations- oder Geschäftsdienstschiicht oder direkt für Clients in der Client-Schicht bereitzustellen. Access Manager stellt beispielsweise die Identity Services für andere Java Enterprise System-Komponenten bereit. |
| Datenschicht            | Bietet Datenbankdienste, auf die andere Dienste innerhalb der Präsentationsschicht oder der Geschäftsdienstschiicht zugreifen können. Directory Server bietet beispielsweise LDAP-Verzeichniszugriff auf andere Dienste.   |

Eine mehrschichtige Architekturstruktur bietet mehrere Vorteile. Während der Phase des Bereitstellungskonzepts können Sie durch die Platzierung der Dienste entsprechend der Funktionalität in einer mehrschichtigen Architektur bestimmen, wie die Dienste in Ihrem Netzwerk verteilt werden sollen. Zudem können Sie feststellen, wie Komponenten innerhalb der Architektur auf Dienste anderer Komponenten zugreifen. Dank dieser Visualisierung können Sie die Verfügbarkeit, Skalierbarkeit, Sicherheit und andere Dienstqualitätslösungen leichter planen.

---

## Beispiel für logische Architekturen

In diesem Abschnitt werden einige Beispiele logischer Architekturen für Java Enterprise System-Lösungen angegeben. In diesen Beispielen werden die Platzierung logischer Komponenten innerhalb der entsprechenden Schichten einer mehrschichtigen Architektur und die anschließende Analyse der Beziehungen zwischen den Komponenten durch Prüfung der Anwendungsfälle dargestellt. Verwenden Sie die Beispiele für logische Architekturen in diesem Abschnitt als Wissensgrundlage für den Entwurf logischer Architekturen in Java Enterprise System-Lösungen.

Beim ersten Beispiel handelt es sich um eine grundlegende Messaging Server-Lösung, die verdeutlicht, wie logisch eindeutige Komponenten von Messaging Server mit anderen Komponenten interagieren. Im zweiten Beispiel wird eine logische Architektur für eine identitätsbasierte Bereitstellungslösung dargestellt, die für ein Unternehmen mittlerer Größe mit etwa 1000 bis 5000 Angestellten geeignet ist.

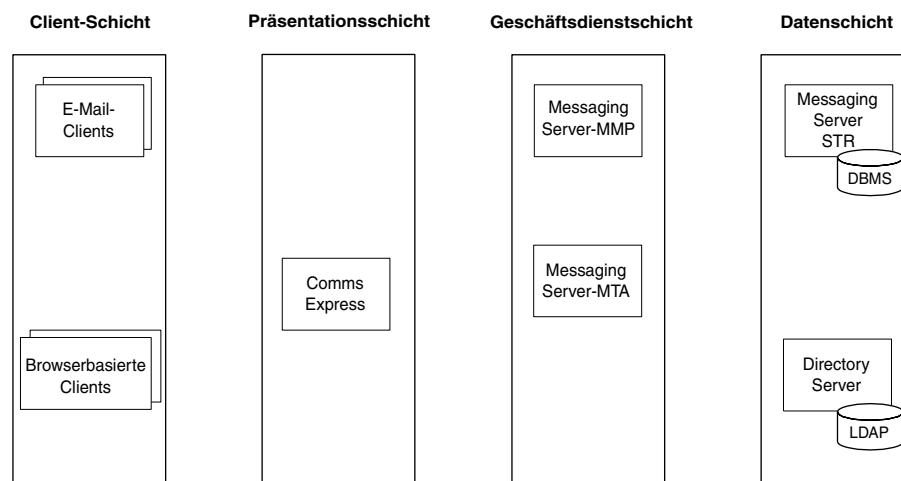
### Messaging Server-Beispiel

In der folgenden Abbildung wird eine grundlegende logische Architektur für die Bereitstellung von Messaging Server dargestellt. In dieser logischen Architektur werden nur die logisch eindeutigen Komponenten angegeben, die für Messaging Server erforderlich sind. In späteren Abbildungen werden die Beziehungen zwischen diesen Komponenten dargestellt.

---

**Hinweis** – In der Regel ist die Bereitstellung von Messaging Server Bestandteil einer Unternehmenslösung, die auch andere Java Enterprise System-Komponenten enthält, wie unter „[Identitätsbasiertes Kommunikationsbeispiel](#)“ auf Seite 63 beschrieben.

---



**ABBILDUNG 4-4** Logische Architektur für die Messaging Server-Bereitstellung

In der folgenden Tabelle werden die in „Messaging Server-Beispiel“ auf Seite 58 dargestellten Komponenten beschrieben.

**TABELLE 4-5** Komponenten in der logischen Architektur von Messaging Server

| Komponente           | Beschreibung  |
|----------------------|---|
| E-Mail-Clients       | Client-Anwendungen zum Lesen und Senden von E-Mails.  |
| Messaging Server MTA | Messaging Server konfiguriert als Message Transfer Agent (MTA) für den Empfang, die Weiterleitung, den Transport und die Zustellung von E-Mail-Nachrichten.   |
| Messaging Server MMP | Messaging Server konfiguriert als Message Multiplexor (MMP) für die Weiterleitung von Verbindungen an entsprechende Nachrichtenspeicher zum Abrufen und Speichern. MMP greift auf Directory Server zu, um Verzeichnisinformationen nachzuschlagen, über die der richtige Nachrichtenspeicher bestimmt wird. |
| Messaging Server STR | Messaging Server konfiguriert als Nachrichtenspeicher zum Abrufen und Speichern von E-Mail-Nachrichten.   |
| Directory Server     | Bietet Zugriff auf LDAP-Verzeichnisdaten.   |

In der logischen Architektur wird die Replikation der Dienste für die Messaging Server-Komponenten nicht angegeben. Bei Unternehmensbereitstellungen werden in der Regel separate MTA-Instanzen für den Eingang und für den Ausgang erstellt, in „Messaging Server-Beispiel“ auf Seite 58 wird jedoch nur eine MTA-Komponente aufgeführt. Die Replikation der logischen Komponenten in mehreren Instanzen ist eine strukturelle Entscheidung, die Sie während der Phase des Bereitstellungs-konzepts treffen.

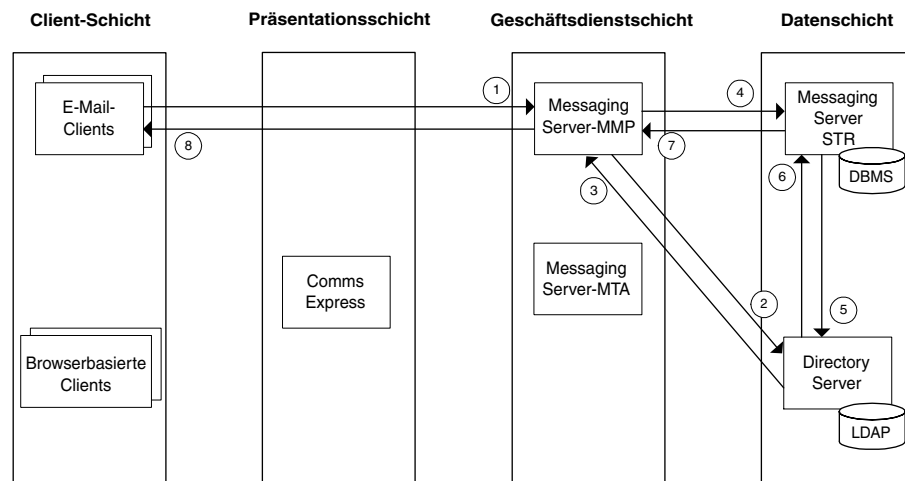
## Messaging Server-Anwendungsfälle

Mithilfe von Anwendungsfällen können Sie Beziehungen zwischen den logischen Komponenten einer Architektur bestimmen. Durch die Zuordnung der Interaktionen zwischen den Komponenten entsprechend den Anwendungsfällen erhalten Sie eine visuelle Darstellung der Komponenteninteraktion, die hilfreich für die Erstellung einer Bereitstellungsstruktur ist.

In der Regel analysieren Sie die einzelnen Anwendungsfälle, um die Interaktion der Komponenten vor der Erstellung der Bereitstellungsstruktur zu bestimmen. Die folgenden drei Anwendungsfälle sind charakteristisch für Messaging Server. Sie zeigen die Interaktionen zwischen den logischen Komponenten.

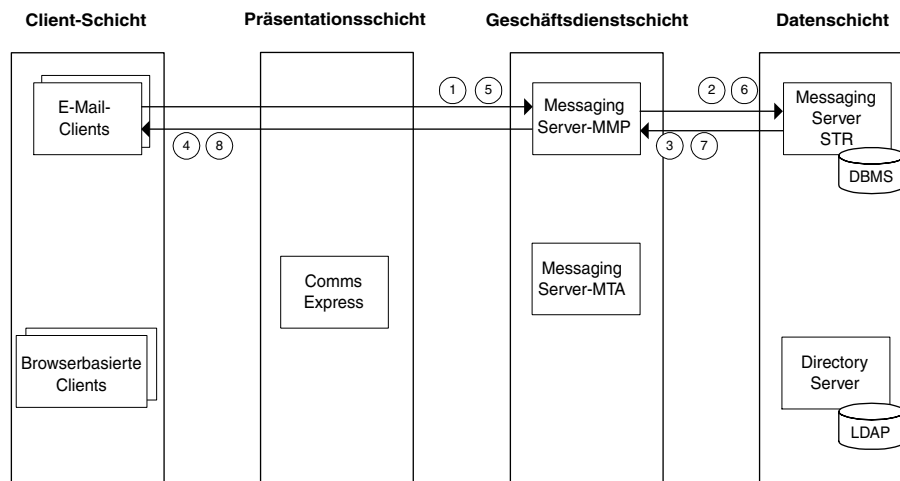
### ▼ Anwendungsfall 1: Ein Benutzer meldet sich erfolgreich bei Messaging Server an

- Schritte**
1. Der E-Mail-Client sendet Anmeldeinformationen an den Messaging Server Multiplexor (MMP).
  2. MMP fordert die Bestätigung der Benutzer-ID und des Passworts von Directory Server an.
  3. Directory Server sendet die Bestätigung an MMP zurück.
  4. MMP fordert eine Nachrichtenliste des Messaging Server-Nachrichtenspeichers (STR) an.
  5. STR fordert den LDAP-Datensatz des Benutzers von Directory Server an.
  6. Directory Server sendet den LDAP-Datensatz des Benutzers an STR zurück.
  7. STR sendet die Nachrichtenliste an MMP zurück.
  8. MMP leitet die Nachrichtenliste an den E-Mail-Client weiter.



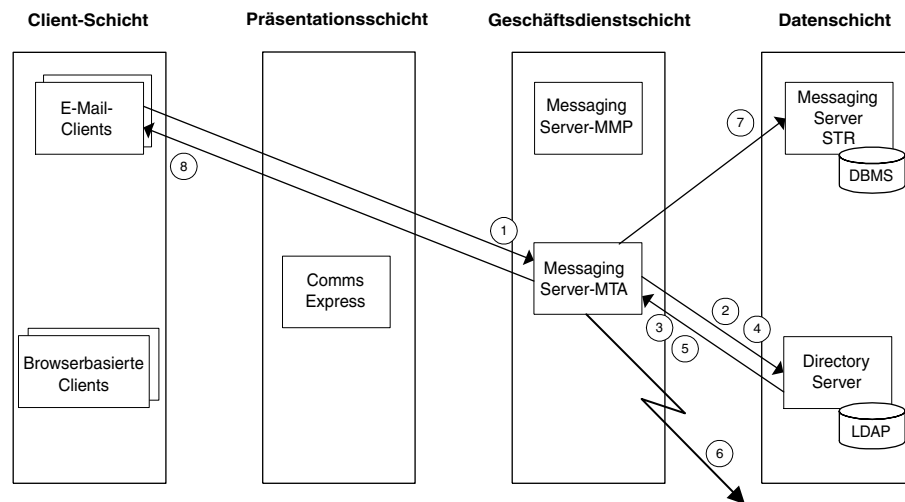
▼ Anwendungsfall 2: Ein angemeldeter Benutzer liest und löscht eine E-Mail

- Schritte**
1. Der E-Mail-Client fordert den Lesevorgang der Meldung aus Messaging Server Multiplexor (MMP) an.
  2. MMP fordert die Nachricht aus dem Messaging Server-Nachrichtenspeicher (STR) an.
  3. STR sendet die Nachricht an MMP zurück.
  4. MMP leitet die Nachricht an den E-Mail-Client weiter.
  5. Der E-Mail-Client sendet den Vorgang zum Löschen der Nachricht an MMP.
  6. MMP leitet den Vorgang zum Löschen der Nachricht an STR weiter.
  7. STR löscht die Nachricht aus der Datenbank und sendet eine entsprechende Bestätigung an MMP.
  8. MMP leitet die Bestätigung des Löschvorgangs an den E-Mail-Client weiter.



### ▼ Anwendungsfall 3: Ein angemeldeter Benutzer sendet eine E-Mail-Nachricht

- Schritte**
1. Der E-Mail-Client sendet eine im Client erstellte Nachricht an den Messaging Server Message Transfer Agent (MTA).
  2. MTA fordert die Bestätigung der Benutzer-ID und des Passworts von Directory Server an.
  3. Directory Server sendet die Bestätigung an MTA zurück.
  4. MTA überprüft die Zieldomäne für alle Empfänger in Directory Server.
  5. Directory Server sendet die Zieldomänen für alle Empfänger an MTA zurück.
  6. MTA leitet die Nachricht an die einzelnen Empfänger weiter.
  7. MTA eitet die Nachricht an den Messaging Server-Nachrichtenspeicher (STR) weiter, um die Nachricht im Postausgang zu speichern.
  8. MTA sendet eine Bestätigung an den E-Mail-Client.



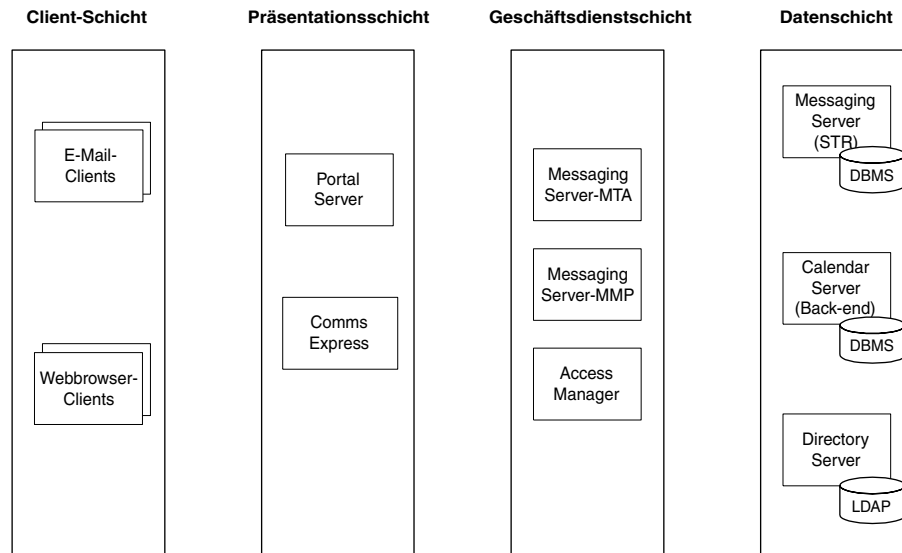
## Identitätsbasiertes Kommunikationsbeispiel

In diesem Beispiel wird eine identitätsbasierte Kommunikationslösung für ein Unternehmen mittlerer Größe mit etwa 1.000 bis 5.000 Mitarbeitern dargestellt. In der Regel ist eine ausführliche Geschäftsanalyse mit anschließender detaillierter Bestimmung der technischen Anforderungen erforderlich, um die logische Architektur zu entwerfen. Da es sich hierbei jedoch um ein theoretisches Beispiel handelt, wird davon ausgegangen, dass die folgenden Geschäftsanforderungen bestimmt wurden:

- Die Mitarbeiter des Unternehmens benötigen persönlichen Zugriff auf interne Websites, Kommunikationsdienste, Kalenderdienste und andere Ressourcen.
- Die unternehmensweite Authentifizierung und Autorisierung bietet Zugriff auf interne Websites und andere Dienste.
- Einzelne Identitäten werden für alle Unternehmensdienste verfolgt. Dies ermöglicht einen Single Sing-On (SSO), welcher den Zugriff auf die internen Websites und andere Dienste ermöglicht.

In den Anwendungsfällen für dieses Beispiel würden die Anmeldeverfahren, das Lesen und Senden von E-Mails, die persönliche Anpassung des Portals, die Synchronisierung von Kalendern und andere ähnliche Benutzeraktivitäten detailliert beschrieben.

In der folgenden Abbildung wird eine logische Architektur für diese Art von identitätsbasierter Kommunikationslösung dargestellt.



## Anwendungsfälle für identitätsbasiertes Kommunikationsbeispiel

Für eine Bereitstellungslösung dieser Art werden in der Regel mehrere detaillierte Anwendungsfälle erstellt, in denen die Interaktion der Benutzer mit den durch die Lösung bereitgestellten Diensten beschrieben wird. Bei diesem Beispiel steht die Interaktion der Komponenten bei der Anmeldung eines Benutzers an einem Portal über einen Webbrowser-Client im Vordergrund. In diesem Beispiel wird das Anmeldungszenario in zwei Anwendungsfälle unterteilt:

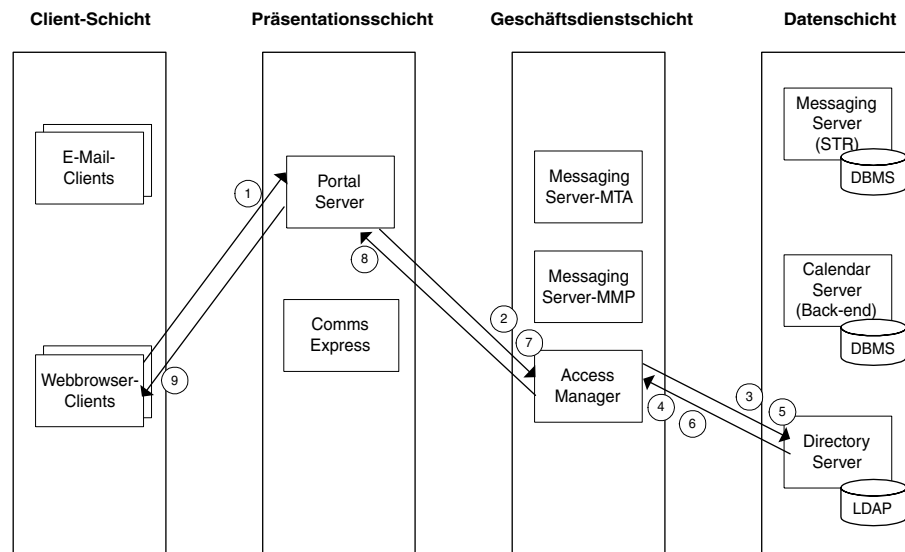
- Der Benutzer meldet sich an, wird authentifiziert und Portal Server ruft die Portalkonfiguration des Benutzers ab.
- Portal Server ruft E-Mail- und Kalenderinformationen zur Anzeige im Web-Client ab.

Die beiden Anwendungsfälle können als ein erweiterter Anwendungsfall betrachtet werden. In diesem Beispiel werden die Anwendungsfälle der Einfachheit halber separat dargestellt.



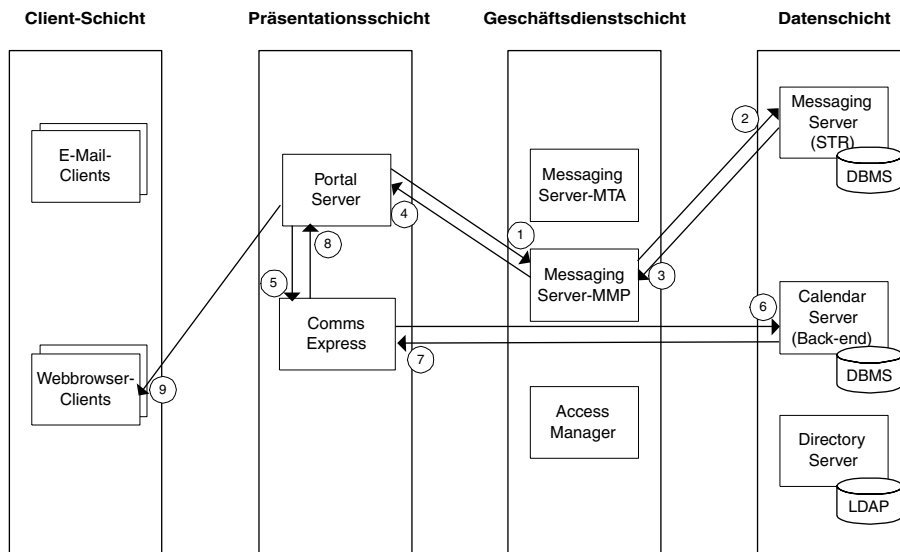
▼ Anwendungsfall 1: Ein Benutzer meldet sich erfolgreich an und das Portal ruft die Konfiguration des Benutzers ab

- Schritte**
1. Der Webbrowser-Client sendet die Benutzer-ID und das Passwort an Portal Server.
  2. Portal Server fordert die Authentifizierung von Access Manager an.
  3. Access Manager fordert die Bestätigung der Benutzer-ID und des Passworts von Directory Server an.
  4. Directory Server überprüft die Benutzer-ID und das Passwort.
  5. Access Manager fordert das Benutzerprofil von Directory Server an.
  6. Directory Server sendet das Benutzerprofil zurück.
  7. Portal Server fordert das Anzeigeprofil des Benutzers von Access Manager an.
  8. Access Manager sendet die Portalkonfiguration zurück.
  9. Die Portalkonfiguration wird im Webbrowser-Client angezeigt.



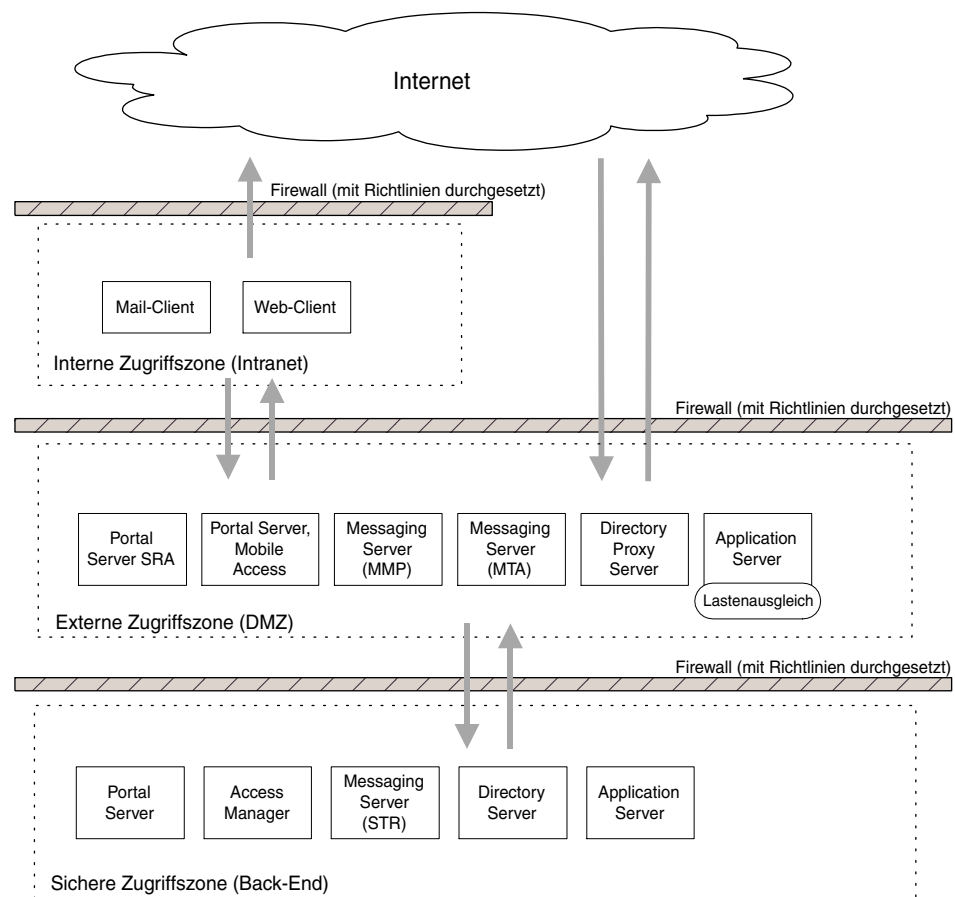
## ▼ Anwendungsfall 2: Portal Server zeigt E-Mail- und Kalenderinformationen an

- Schritte**
1. Nach erfolgter Anmeldung, Authentifizierung und dem Abruf der Portalkonfiguration fordert Portal Server E-Mail-Nachrichten von Messaging Server MMP an.
  2. MMP fordert eine Nachrichtenliste von Messaging Server STR an.
  3. STR sendet die Nachrichtenliste an MMP zurück.
  4. MMP leitet die Nachrichten-Header an Portal Server weiter.
  5. Portal Server fordert Kalenderinformationen von Communications Express an.
  6. Communications Express fordert Kalenderinformationen vom Calendar Server-Back-End an.
  7. Calendar Server-Back-End sendet Kalenderinformationen an Communications Express zurück.
  8. Communications Express leitet Kalenderinformationen an Portal Server weiter.
  9. Portal Server sendet alle Kanalinformationen an den Webbrowser-Client.



# Zugriffszonen

Eine weitere Möglichkeit, die Komponenten einer logischen Architektur darzustellen, ist deren Platzierung in Zugriffszonen, die anzeigen, auf welche Weise die Architektur einen sicheren Zugriff bereitstellt. In der folgenden Abbildung werden die Zugriffszonen für die Bereitstellung der Java Enterprise System-Komponenten dargestellt. Für jede Zugriffszone wird angezeigt, auf welche Weise die Komponenten einen sicheren Remote-Zugriff auf und über das Internet bzw. Intranet bieten.



**ABBILDUNG 4-5** In Zugriffszonen platzierte logische Komponenten

In der folgenden Tabelle werden die in der Abbildung „Zugriffszonen“ auf Seite 67 dargestellten Zugriffszonen beschrieben.

TABELLE 4-6 Sichere Zugriffszonen und darin platzierte Komponenten

| Zugriffszone                    | Beschreibung   |
|---------------------------------|--|
| Interne Zugriffszone (Intranet) | <p>Der Internetzugang über Richtlinien, die mit einer Firewall zwischen Intranet und Internet durchgesetzt werden. Die interne Zugriffszone wird in der Regel von Endbenutzern zum Browsen im Internet und zum Senden von E-Mails verwendet.</p> <p>In einigen Fällen ist der direkte Zugriff auf das Internet zum Browsen zulässig. In der Regel wird jedoch der sichere Zugriff auf das und aus dem Internet über die externe Zugriffszone ermöglicht.</p> |
| Externe Zugriffszone (DMZ)      | Bietet sicheren Zugriff auf das und aus dem Internet und fungiert als Sicherheitspuffer für wichtige Back-End-Dienste.   |
| Sichere Zugriffszone (Back-End) | Bietet eingeschränkten Zugriff auf wichtige Back-End-Dienste, auf die ausschließlich von einer externen Zugriffszone aus zugegriffen werden kann.  |

In der Abbildung „Zugriffszonen“ auf Seite 67 sind die in den vorherigen Beispielen aufgeführten logischen Schichten nicht dargestellt, sondern im Mittelpunkt stehen die Komponenten, die den Remote-Zugriff und den internen Zugriff ermöglichen, die Beziehung dieser Komponenten zu Sicherheitsmaßnahmen, wie Firewalls und eine bildliche Darstellung der umzusetzenden Zugriffsregeln. Verwenden Sie die mehrschichtige Architekturstruktur in Kombination mit der Struktur, in der die Zugriffszonen dargestellt werden, um ein logisches Modell Ihrer geplanten Bereitstellung zu erstellen.

## Bereitstellungsszenario

Der fertige Entwurf der logischen Architektur allein reicht nicht aus, um zur Phase der Bereitstellungsstruktur im Lösungszyklus überzugehen. Sie müssen die logische Architektur mit den Dienstqualitätsanforderungen (QoS) kombinieren, die in der Phase der technischen Anforderungen bestimmt wurden. Die Kombination der logischen Architektur und der QoS-Anforderungen stellt ein Bereitstellungszenario dar. Das Bereitstellungszenario ist der Ausgangspunkt für den Entwurf der Bereitstellungsarchitektur, wie in Kapitel 5 beschrieben.

## Bereitstellungskonzept

---

In der Entwicklungsphase eines Bereitstellungskonzepts während des Lebenszyklus der Lösung entwerfen Sie eine Bereitstellungsarchitektur auf hoher Ebene und eine Implementierungsspezifikation auf niedriger Ebene und bereiten eine Reihe von Plänen und Spezifikationen vor, die für die Implementierung der Lösung erforderlich sind. Die Projektgenehmigung findet in der Bereitstellungskonzeptphase statt.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Informationen zu Bereitstellungskonzepten“ auf Seite 69
- „Methodik des Bereitstellungskonzepts“ auf Seite 73
- „Einschätzen von Prozessoranforderungen“ auf Seite 74
- „Einschätzen von Prozessoranforderungen für sichere Transaktionen“ auf Seite 81
- „Ermitteln von Verfügbarkeitsstrategien“ auf Seite 84
- „Festlegen von Strategien hinsichtlich der Skalierbarkeit“ auf Seite 93
- „Identifizieren von Leistungsgaps“ auf Seite 95
- „Konzeption für optimale Ressourcennutzung“ auf Seite 98
- „Verwalten von Risiken“ auf Seite 99
- „Beispiel für eine Bereitstellungsarchitektur“ auf Seite 100

---

## Informationen zu Bereitstellungskonzepten

Die Erstellung eines Bereitstellungskonzepts beginnt mit dem Bereitstellungsszenario, das während der Phasen des logischen Konzepts und der technischen Anforderungen des Lösungslebenszyklus erstellt wurde. Das Bereitstellungsszenario umfasst eine logische Architektur und die für die Lösung erforderlichen Dienstqualitätsanforderungen (Quality of Service, QoS). Die in der logischen Architektur identifizierten Komponenten werden für physische Server und andere Netzwerkgeräte zugeordnet, um eine Bereitstellungsarchitektur zu erstellen. Die

QoS-Anforderungen dienen als Anhaltspunkte hinsichtlich der Hardwarekonfigurationen in Bezug auf Leistung, Verfügbarkeit, Skalierbarkeit und andere QoS-Spezifikationen in diesem Zusammenhang.

Die Konzipierung der Bereitstellungsarchitektur ist ein von Wiederholungen geprägter Vorgang. Die QoS-Anforderungen werden üblicherweise erneut zurate gezogen und die im Vorfeld festgelegten Konzepte nochmals überprüft. Dabei werden die Wechselbeziehungen der QoS-Anforderungen berücksichtigt und es wird versucht, ein Gleichgewicht zwischen dem so genannten Tradeoff und den Gesamtkosten zu erzielen, um so letztlich über eine optimale Lösung zu verfügen, die den Geschäftszielen des Projekts gerecht wird.

## Projektgenehmigung

Die Projektgenehmigung erfolgt während der Bereitstellungs-konzeptphase, üblicherweise nach der Erstellung der Bereitstellungsarchitektur. Die geschätzten realen Bereitstellungskosten werden anhand der Bereitstellungsarchitektur und gegebenenfalls anhand der nachfolgend beschriebenen Implementierungsspezifikationen ermittelt und den Interessengruppen zur Genehmigung vorgelegt. Im Anschluss an die Projektgenehmigung werden Verträge hinsichtlich des Bereitstellungsabschlusses unterzeichnet und es werden die Ressourcen erworben und zugeordnet, die zur Implementierung des Projekts erforderlich sind.

## Bereitstellungskonzeptausgaben

In der Bereitstellungs-konzeptphase werden möglicherweise folgende Spezifikationen und Pläne ausgearbeitet:

- **Bereitstellungsarchitektur.** Eine Architektur auf hoher Ebene, aus der die Zuordnung einer logischen Architektur zu einer physischen Umgebung hervorgeht. Die physische Umgebung umfasst die Computerknoten in einer Intranet- bzw. Internetumgebung, Prozessoren, Speicher, Speichergeräte sowie weitere Hardware und Netzwerkgeräte.
- **Implementierungsspezifikationen.** Detaillierte Spezifikationen, die zur Erstellung der Bereitstellung herangezogen werden. Aus diesen Spezifikationen geht hervor, welche Computer- und Netzwerkhardware erforderlich ist. Außerdem wird hier das Netzwerklayout für die Bereitstellung erläutert. Zu den Implementierungsspezifikationen zählen Directory Services (Verzeichnisdienste), einschließlich Details zu Spezifikationen für einen Verzeichnisinformationsbaum (Directory Information Tree, DIT) sowie der für den Verzeichniszugriff definierten Gruppen und Rollen.
- **Implementierungspläne.** Eine Reihe von Plänen, die unterschiedliche Aspekte der Implementierung einer Unternehmenssoftwarelösung abdecken. Zu den Implementierungsplänen zählen:

- *Migrationsplan*. Erläutert die Strategien und Vorgänge zur Migration von Unternehmensdaten und zur Aufrüstung von Unternehmenssoftware. Die migrierten Daten müssen den Formaten und Standards der neu installierten Unternehmensanwendungen entsprechen. Voraussetzung für die Interoperabilität ist, dass von sämtlicher Unternehmenssoftware die richtige Version vorliegt.
- *Installationsplan*. Dieser Plan wird von der Bereitstellungsarchitektur abgeleitet und gibt Servernamen, Installationsverzeichnis, Installationsreihenfolge, Installationstypen für die einzelnen Knoten sowie die Konfigurationsinformationen an, die für die Installation und Konfiguration einer verteilten Bereitstellung erforderlich sind.
- *Benutzerverwaltungsplan*. Umfasst Migrationsstrategien für Daten in bestehenden Verzeichnissen und Datenbanken, Spezifikationen zum Verzeichniskonzept, die das in der Bereitstellungsarchitektur angegebene Kontoreplikationskonzept berücksichtigen, sowie Vorgehensweisen zur Bereitstellung neuer Inhalte für Verzeichnisse.
- *Testplan*. Beschreibt Verfahren zum Testen der bereitgestellten Software, einschließlich spezifischer Pläne für die Bereitstellung von Prototyp- und Pilotimplementierungen, Belastungstests, bei denen die Fähigkeit zur Handhabung der geplanten Auslastung ermittelt wird sowie Funktionstests, mit denen überprüft wird, ob der Betrieb erwartungsgemäß verläuft.
- *Roll-out-Plan*. Beschreibt die Vorgehensweisen sowie den Terminplan für die Implementierung aus einer Plan- und Testumgebung in eine Produktionsumgebung. Im Normalfall erfolgt die Implementierung in eine Produktionsumgebung in mehreren Phasen. Die erste Phase kann beispielsweise die Bereitstellung der Software für eine kleine Gruppe von Benutzern sein. Dann wird mit jeder weiteren Phase die Benutzeranzahl erhöht, bis die Bereitstellung komplett abgeschlossen ist. Bei der phasenweisen Implementierung kann auch die zeitlich festgelegte Implementierung bestimmter Softwarepakete erfolgen, bis die Bereitstellung komplett abgeschlossen ist.
- *Wiederherstellungsplan*. Beschreibt, wie das System nach unerwarteten Gesamtausfällen wiederhergestellt werden kann. Der Wiederherstellungsplan sieht sowohl Vorgehensweisen für Ausfälle großen als auch kleinen Ausmaßes vor.
- *Betriebsplan (Run Book (Ausführungsbuch))*. Ein Handbuch mit Vorgängen, in denen Überwachungs-, Wartungs-, Installations- und Aufrüstungsverfahren beschrieben werden.
- *Schulungsbuch*. Enthält Vorgänge und Vorgehensweisen zur Schulung von Administratoren und Endbenutzern für die Verwendung der neu installierten Software.

## Faktoren mit Auswirkung auf das Bereitstellungskonzept

Die Entscheidungen, die Sie in der Bereitstellungskonzeptphase treffen, werden von mehreren Faktoren beeinflusst. Berücksichtigen Sie folgende Schlüsselfaktoren:

- **Logische Architektur.** In der logischen Architektur werden die Dienste in einer geplanten Lösung detailliert erläutert; außerdem wird auf die Wechselbeziehung zwischen den Komponenten eingegangen, die diese Dienste zur Verfügung stellen. Verwenden Sie die logische Architektur als Anhaltspunkt dafür, wie Dienste optimal verteilt werden können. Ein Bereitstellungsszenario umfasst die logische Architektur sowie die Anforderungen hinsichtlich der Dienstqualität (siehe nachfolgende Beschreibung).
- **Dienstqualitätsanforderungen.** Anhand der Anforderungen an die Dienstqualität (Quality of Service, QoS) werden unterschiedliche Aspekte des Betriebs einer Lösung angegeben. Verwenden Sie die QoS-Anforderungen, um Strategien auszuarbeiten, mit denen Leistung, Verfügbarkeit, Skalierbarkeit, Wartungseignung sowie andere Ziele in Bezug auf die Dienstqualität erreicht werden können. Ein Bereitstellungsszenario umfasst die (zuvor beschriebene) logische Architektur sowie die Anforderungen hinsichtlich der Dienstqualität.
- **Anwendungsanalyse.** Die Anwendungsanalyse, die in der Phase der technischen Anforderungen im Lösungszyklus erstellt wurde, gibt Aufschluss über Anwendungsmuster, mit deren Hilfe sich Auslastung und Belastung in einem bereitgestellten System einschätzen lassen. Nutzen Sie die Anwendungsanalyse, um leistungsbezogene Engpässe zu isolieren und Strategien zur Erfüllung von QoS-Anforderungen auszuarbeiten.
- **Anwendungsfälle.** In Anwendungsfällen, die in der Phase der technischen Anforderungen im Lösungslebenszyklus erstellt wurden, werden eindeutige Benutzerinteraktionen aufgeführt, die für eine Bereitstellung identifiziert wurden. Oft werden hierbei die gängigsten Anwendungsfälle identifiziert. Obwohl die Anwendungsfälle Bestandteil der Anwendungsanalyse sind, empfiehlt es sich, bei der Bewertung eines Bereitstellungskonzepts die Anwendungsfälle zurate zu ziehen, um sicherzustellen, dass die entsprechenden Probleme richtig angegangen werden.
- **Vereinbarungen auf Dienstebene.** In einer Vereinbarung auf Dienstebene (Service Level Agreement, SLA) werden die leistungsbezogenen Mindestanforderungen angegeben. Hieraus geht ebenfalls hervor, auf welcher Ebene und in welchem Umfang Kundenunterstützung vonnöten ist, wenn diese Anforderungen nicht erfüllt werden. Die in einer Vereinbarung auf Dienstebene angegebenen Leistungsanforderungen sollten von einem Bereitstellungskonzept problemlos erfüllt werden können.
- **Gesamtkosten.** Bei der Ausarbeitung des Bereitstellungskonzepts analysieren Sie potenzielle Lösungen, die sich u. a. auf die Verfügbarkeits-, Leistungs- und Skalierungsaspekte der QoS-Anforderungen beziehen. Bei jeder in Betracht gezogenen Lösung müssen jedoch auch die Kosten dieser Lösung sowie die Auswirkung dieser Lösung auf die Gesamtkosten berücksichtigt werden. Ziehen



Sie in jedem Fall den Tradeoff Ihrer Beziehung in Betracht und stellen Sie sicher, dass Ihre Ressourcen so optimiert wurden, dass die Geschäftsanforderungen innerhalb der Unternehmensgrenzen erfüllt werden.

- **Geschäftsziele.** Geschäftsziele werden während der Phase der Geschäftsanalyse des Lösungszyklus festgelegt; hierzu zählen die Geschäftsanforderungen sowie die unternehmerischen Einschränkungen, die bei der Erfüllung dieser Ziele beachtet werden müssen. Das Bereitstellungskonzept wird letztlich anhand seiner Fähigkeit zur Erfüllung der Geschäftsziele beurteilt.

---

## Methodik des Bereitstellungskonzepts

Genau wie bei anderen Aspekten der Bereitstellungsplanung ist die Ausarbeitung des Bereitstellungskonzepts sowohl eine Kunst als auch eine Wissenschaft; folglich können hierfür keine bis ins kleinste Detail spezifischen Vorgänge und Vorgehensweise vorgegeben werden. Die Faktoren, die zu einem erfolgreichen Bereitstellungskonzept beitragen, gehen über Erfahrung bei der Konzeption, Kenntnisse im Bereich Systemarchitektur und Domänen sowie angewandtes kreatives Denken hinaus.

Bei der Bereitstellungs-konzeption steht im Mittelpunkt, Leistungsvorgaben zu erfüllen und gleichzeitig anderen QoS-Anforderungen gerecht zu werden. Mit den von Ihnen eingesetzten Strategien muss ein Gleichgewicht der Tradeoffs Ihrer Konzeptentscheidungen erreicht werden, um die Lösung zu optimieren. Die verwendete Methodik umfasst im Regelfall folgende Aufgaben:

- **Einschätzen von Prozessoranforderungen.** Der erste Schritt bei der Bereitstellungs-konzeption besteht häufig darin, die Anzahl der Prozessoren (Central Processing Units, CPUs) einzuschätzen, die für die einzelnen Komponenten in der logischen Architektur erforderlich sind. Beginnen Sie mit den Fällen, die für die höchste Auslastung stehen, und gehen Sie dann die restlichen Anwendungsfälle durch. Ziehen Sie die Auslastung sämtlicher Komponenten in Betracht, die die Anwendungsfälle unterstützen, und passen Sie Ihre Einschätzungen entsprechend an. Greifen Sie auch auf Erfahrungen zurück, die Sie bisher bei der Konzeption von Unternehmenssystemen gesammelt haben.
- **Einschätzen von Prozessoranforderungen für sicheren Transport.** Befassen Sie sich mit den Fällen, bei denen sicherer Transport erforderlich ist, und passen Sie die CPU-Einschätzungen entsprechend an.
- **Replizieren von Diensten hinsichtlich Verfügbarkeit und Skalierbarkeit.** Wenn die Prozessoreinschätzungen Ihren Vorstellungen entsprechen, nehmen Sie die Änderungen am Konzept vor, die hinsichtlich der QoS-Anforderungen in Bezug auf Verfügbarkeit und Skalierbarkeit erforderlich sind. Ziehen Sie hierbei Lösungen für den Lastenausgleich in Betracht, bei denen Verfügbarkeitsaspekte sowie Failover-Faktoren angegangen werden.

Beachten Sie bei der Analyse die Tradeoffs Ihrer Konzeptentscheidungen. So stellen sich beispielsweise folgende Fragen: Welche Auswirkungen hat die Verfügbarkeits- und Skalierbarkeitsstrategie auf die Wartungseignung des Systems? Welche

sonstigen Kosten entstehen durch die Strategien?

- **Identifizieren von Engpässen..** Überprüfen Sie bei weiteren Analyseschritten das Bereitstellungskonzept, um mögliche Engpässe auszumachen, durch die die Datenübertragungsrate unter die Anforderungsgrenze absinkt, und nehmen Sie die entsprechenden Anpassungen vor.
- **Optimieren von Ressourcen.** Überprüfen Sie Ihr Bereitstellungskonzept in Bezug auf die Ressourcenverwaltung und ziehen Sie hierbei Optionen in Betracht, die die Anforderungen erfüllen und gleichzeitig die Kosten senken.
- **Verwalten von Risiken.** Sehen Sie sich Ihre geschäftsbezogenen und technischen Analysen in Bezug auf Ihr Konzept nochmals durch und nehmen Sie Änderungen vor, die Ereignisse oder Situationen abdecken, die in früheren Planungsphasen außer Acht gelassen wurden.

---

## Einschätzen von Prozessoranforderungen

In diesem Abschnitt wird erläutert, wie die Anzahl der Prozessoren (Central Processing Units, CPUs) sowie der erforderliche Speicherbedarf zur Unterstützung der Dienste in einem Bereitstellungskonzept ermittelt werden können. Anhand eines Beispielszenarios einer Kommunikationsbereitstellung wird Schritt für Schritt erklärt, wie die Einschätzung in diesem Fall abläuft.

Bei der Einschätzung der CPU-Kapazität handelt es sich um einen von Wiederholungen geprägten Vorgang, bei dem folgende Punkte beachtet werden:

- Logische Komponenten und deren Interaktion (gemäß der Darstellung von Komponentenabhängigkeiten in der logischen Architektur)
- Anwendungsanalyse für die identifizierten Anwendungsfälle
- Dienstqualitätsanforderungen
- Bisher gesammelte Erfahrungen mit Bereitstellungskonzepten und Java Enterprise System
- Beratung mit Sun-Anbietern professioneller Dienste, die auf dem Gebiet der Konzeption und Implementierung unterschiedlicher Bereitstellungstypen erfahren sind

Der Einschätzungsvorgang umfasst die nachfolgend aufgeführten Schritte. Die Reihenfolge dieser Schritte ist nicht zwingend vorgegeben, stellt jedoch eine Möglichkeit zum Umgang mit den Faktoren dar, die sich auf das Endergebnis auswirken.

1. Ermitteln Sie eine CPU-Basiseinschätzung für Komponenten, die für das System als Benutzereinstiegspunkte identifiziert wurden.

Eine bei der Konzeption zu treffende Entscheidung besteht darin, festzulegen, ob CPUs vollständig oder teilweise geladen werden. Durch vollständig geladene CPUs wird die Kapazität eines Systems maximiert. Bei der Kapazitätssteigerung fallen Wartungskosten an; außerdem muss die mögliche Ausfallzeit beachtet werden, die durch das Hinzufügen weiterer CPUs anfällt. In einigen Fällen haben Sie die Möglichkeit, weitere Computer hinzuzufügen, um den wachsenden Leistungsanforderungen gerecht zu werden.

Durch teilweise geladene CPUs wird die Verarbeitung übermäßiger Leistungsanforderungen ermöglicht, ohne dass hierbei sofort Wartungskosten anfallen. Aufgrund des nicht voll ausgelasteten Systems fallen jedoch Vorabkosten an.

2. Passen Sie die CPU-Einschätzungen so an, dass die Interaktion zwischen Komponenten berücksichtigt wird.

Befassen Sie sich eingehend mit der Interaktion zwischen Komponenten in der logischen Architektur, um zu ermitteln, in welchen Fällen es durch abhängige Komponenten zu zusätzlicher Auslastung kommt.

3. Überprüfen Sie die Anwendungsanalyse auf spezifische Anwendungsfälle hin, um die Spitzenauslastung für das System zu ermitteln, und nehmen Sie dann Änderungen an den Komponenten vor, die zur Verarbeitung dieser Spitzenauslastung vorgesehen sind.

Beginnen Sie mit den Anwendungsfällen mit der größten Gewichtung (die zur größten Auslastung führen) und gehen Sie dann alle verbleibenden Anwendungsfälle durch, um zu gewährleisten, dass sämtliche geplanten Anwendungsszenarios abgedeckt sind.

4. Passen Sie die CPU-Einschätzungen so an, dass die Sicherheits-, Verfügbarkeits- und Skalierbarkeitsanforderungen berücksichtigt werden.

Durch den Einschätzungsvorgang werden Anhaltspunkte zur Ermittlung der tatsächlich benötigten Verarbeitungsleistung bereitgestellt. Im Normalfall werden Prototypbereitstellungen basierend auf diesen Einschätzungen erstellt; anschließend erfolgt das strenge Testverfahren anhand erwarteter Anwendungsfälle. Die tatsächlichen Verarbeitungsanforderungen für ein Bereitstellungs-konzept lassen sich nur durch wiederholte Testläufe ermitteln.

## Beispiel für das Einschätzen von Prozessoranforderungen

In diesem Abschnitt wird eine der Methoden erläutert, mit der die für eine Beispielbereitstellung erforderliche Verarbeitungsleistung eingeschätzt werden kann. Die Beispielbereitstellung basiert auf der logischen Architektur der identitätsbasierten Kommunikationslösung für ein mittelständisches Unternehmen (1000 bis 5000 Mitarbeiter), die im Abschnitt „[Identitätsbasiertes Kommunikationsbeispiel](#)“ auf Seite 63 erläutert wird.

Die in diesem Beispiel verwendeten CPU-Werte und Speicherangaben sind willkürlich gewählt und dienen lediglich der Veranschaulichung. Diese Angaben sind auf willkürlich gewählten Daten begründet, auf denen das theoretische Beispiel basiert. Zur Einschätzung der Prozessoranforderungen ist die eingehende Analyse unterschiedlicher Faktoren erforderlich. Diese Analyse würde folgende Informationen umfassen, ist jedoch nicht auf sie beschränkt:

- Detaillierte Anwendungsfälle und Anwendungsanalyse basierend auf eingehender Geschäftsanalyse
- Durch Analyse von Geschäftsanforderungen ermittelte QoS-Anforderungen
- Spezifische Kosten und Spezifikationen bezüglich Verarbeitungs- und Netzwerkhardware
- Bei der Implementierung vergleichbarer Bereitstellungen gesammelte Erfahrungen



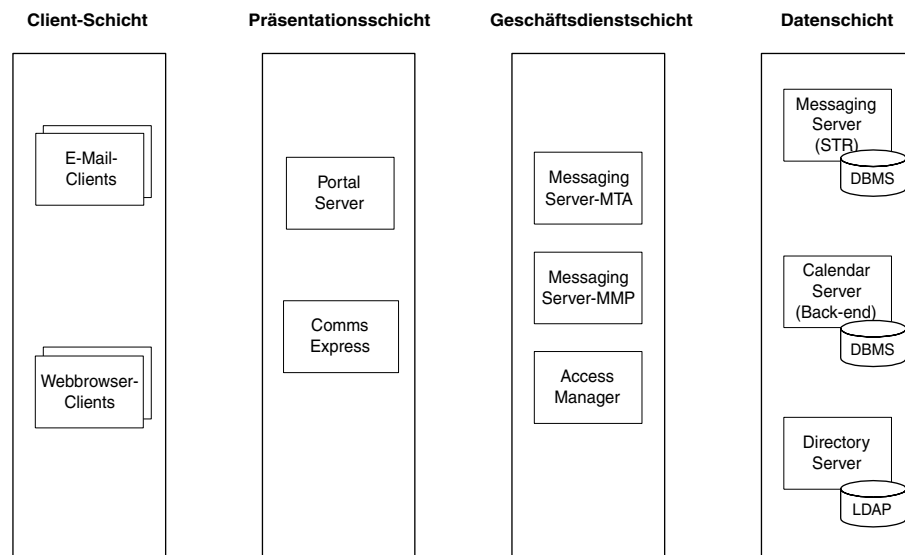
---

**Achtung** – Die in diesen Beispielen enthaltenen Informationen dienen nicht als spezifische Implementierungshinweise, sondern lediglich der Veranschaulichung eines Vorgangs, den Sie bei der Konzeption eines Systems möglicherweise durchführen.

---

## CPU-Basiseinschätzung für Benutzereinstiegspunkte ermitteln

Schätzen Sie zunächst die Anzahl an CPUs ab, die zur Verarbeitung der erwarteten Auslastung sämtlicher Komponenten benötigt werden, bei denen es sich um Benutzereinstiegspunkte handelt. Die folgende Abbildung zeigt die logische Architektur für ein identitätsbasiertes Kommunikationsszenario, das zuvor unter „[Identitätsbasiertes Kommunikationsbeispiel](#)“ auf Seite 63 beschrieben wurde.



**ABBILDUNG 5–1** Logische Architektur für ein identitätsbasiertes Kommunikationsszenario

In der folgenden Tabelle sind die Komponenten in der Präsentationsschicht der logischen Architektur aufgeführt, die in direkter Verbindung mit dem Endbenutzer der Bereitstellung stehen. Die Tabelle enthält CPU-Basiseinschätzungen, die von der Analyse der technischen Anforderungen, von Anwendungsfällen, von der spezifischen Anwendungsanalyse sowie von mit diesem Bereitstellungstyp gesammelten Erfahrungen abgeleitet wurden.

**TABELLE 5–1** CPU-Einschätzungen für Komponenten mit zugriffsbezogenen Benutzereinstiegspunkten

| Komponente             | Anzahl der CPUs | Beschreibung  |
|------------------------|-----------------|---|
| Portal Server          | 4               | Komponente, die ein Benutzereinstiegspunkt ist.                         |
| Communications Express | 2               | Leitet Daten an Messaging- und Kalenderkanäle von Portal Server weiter. |

## Aufnehmen der CPU-Einschätzungen für Dienstabhängigkeiten

Für die Komponenten, die Benutzereinstiegspunkte zur Verfügung stellen, ist die Unterstützung weiterer Java Enterprise System-Komponenten erforderlich. Wenn Sie die Angabe von Leistungsanforderungen fortsetzen möchten, nehmen Sie die Leistungseinschätzungen auf, damit die von anderen Komponenten benötigte Unterstützung berücksichtigt wird. Bei der Konzeption der logischen Architektur sollte der Typ der Interaktion zwischen Komponenten genau angegeben werden, wie im Beispiel mit der logischen Architektur im Abschnitt „Beispiel für logische Architekturen“ auf Seite 58 erläutert.

**TABELLE 5–2** CPU-Einschätzungen für unterstützende Komponenten

| Komponente                                 | CPUs | Beschreibung   |
|--|------|--|
| Messaging Server MTA (Eingang)             | 1    | Leitet eingehende Mail-Nachrichten von Communications Express und E-Mail-Clients weiter.                                     |
| Messaging Server MTA (Ausgang)             | 1    | Leitet ausgehende Mail-Nachrichten an die Empfänger weiter.  |
| Messaging Server MMP                       | 1    | Ermöglicht den Zugriff auf Messaging Server-Nachrichtenspeicher für E-Mail-Clients.  |
| Messaging Server STR (Nachrichtenspeicher) | 1    | Ruft E-Mail-Nachrichten ab und speichert sie.  |
| Access Manager                             | 2    | Stellt Autorisierungs- und Authentifizierungsdienste bereit.   |
| Calendar Server (Back-End)                 | 2    | Ruft Kalenderdaten für Communications Express, ein Calendar Server-Front-End, ab und speichert sie.                          |
| Directory Server                           | 2    | Stellt LDAP-(Lightweight Directory Access Protocol-)Verzeichnisdienste zur Verfügung.  |
| Web Server                                 | 0    | Bietet Webcontainer-Unterstützung für Portal Server und Access Manager.<br><br>(Keine zusätzlichen CPU-Zyklen erforderlich.) |

## Anwendungsfälle auf Spitzenauslastung hin prüfen

Kehren Sie zu den Anwendungsfällen und der Anwendungsanalyse zurück, um die Bereiche zu identifizieren, in denen es zur Spitzenauslastung kommt, und passen Sie Ihre CPU-Einschätzungen entsprechend an.

In diesem Beispiel wird angenommen, dass folgende Spitzenladebedingungen identifiziert werden:

- Anfänglicher stufenweiser Anstieg von Benutzern bei der gleichzeitigen Anmeldung
- E-Mail-Austausch innerhalb vorgegebener Zeitrahmen

Um diese Spitzenauslastung zu berücksichtigen, nehmen Sie Änderungen an den Komponenten vor, von denen diese Dienste zur Verfügung gestellt werden. In der nachfolgenden Tabelle werden die Anpassungen erläutert, die vorgenommen werden können, um diese Spitzenauslastung zu berücksichtigen.

**TABELLE 5-3** Anpassungen der CPU-Einschätzung hinsichtlich der Spitzenauslastung

| Komponente                                 | CPUs (angepasst) | Beschreibung                                       |
|--|------------------|--|
| Messaging Server MTA (Eingang)             | 2                | 1 CPU für eingehende E-Mails (Spitze) hinzufügen   |
| Messaging Server MTA (Ausgang)             | 2                | 1 CPU für ausgehende E-Mails (Spitze) hinzufügen   |
| Messaging Server MMP                       | 2                | 1 CPU für zusätzliche Auslastung hinzufügen        |
| Messaging Server STR (Nachrichtenspeicher) | 2                | 1 CPU für zusätzliche Auslastung hinzufügen        |
| Directory Server                           | 3                | 1 CPU für zusätzliche LDAP-Suchvorgänge hinzufügen |

## Einschätzungen für andere Ladebedingungen ändern

Fahren Sie mit Ihren CPU-Einschätzungen fort, um weitere QoS-Anforderungen zu berücksichtigen, die sich auf die Auslastung auswirken können:

- **Sicherheit.** Ermitteln Sie in Phase der technischen Anforderungen, wie sich der sichere Transport von Daten auf die Auslastungsanforderungen auswirken kann, und passen Sie Ihre Einschätzungen entsprechend an. Im folgenden Abschnitt, „[Einschätzen von Prozessoranforderungen für sichere Transaktionen](#)“ auf Seite 81, wird erläutert, wie die Anpassungen vorgenommen werden können.
- **Replikation von Diensten.** Passen Sie die CPU-Einschätzungen dahin gehend an, dass die Replikation von Diensten bezüglich Verfügbarkeit, Lastenausgleich und Skalierbarkeit berücksichtigt werden. Im folgenden Abschnitt, „[Ermitteln von Verfügbarkeitsstrategien](#)“ auf Seite 84, werden Größenaspekte für Verfügbarkeitslösungen erläutert. Im Abschnitt „[Festlegen von Strategien hinsichtlich der Skalierbarkeit](#)“ auf Seite 93 werden Lösungen hinsichtlich des verfügbaren Zugriffs auf Verzeichnisdienste erläutert.
- **Latente Kapazität und Skalierbarkeit.** Ändern Sie CPU-Einschätzungen dahin gehend, dass für unerwartet hohe Auslastungen in der Bereitstellung latente Kapazität genutzt werden kann. Überprüfen Sie die erwarteten Meilensteine für Skalierung und geplanten Lastenanstieg im Laufe der Zeit, um zu gewährleisten,

dass jeder geplante Meilenstein zur Skalierung des Systems (horizontal oder vertikal) erreicht werden kann.

## CPU-Einschätzungen aktualisieren

Im Normalfall werden CPUs auf eine gerade Zahl aufgerundet. Durch das Aufrunden auf eine gerade Zahl können die CPU-Einschätzungen zu gleichen Teilen zwischen zwei physischen Servern aufgeteilt werden; zudem wird ein kleiner Faktor für latente Kapazität hinzugefügt. Sie sollten beim Aufrunden jedoch Ihre spezifischen Anforderungen hinsichtlich der Replikation von Diensten berücksichtigen.

Als Faustregel werden für jede CPU 2 GB Speicher vorgesehen. Wie viel Speicher tatsächlich benötigt wird, hängt von Ihren spezifischen Gegebenheiten ab und kann durch Tests ermittelt werden.

In der folgenden Tabelle sind die endgültigen Einschätzungen für das identitätsbasierte Kommunikationsbeispiel aufgeführt. In diesen Einschätzungen ist keinerlei zusätzliche Rechnerkapazität berücksichtigt, die für erhöhte Sicherheit und größere Verfügbarkeit hätte hinzugefügt werden können. Die Gesamtkapazität für Sicherheit und Verfügbarkeit wird in den nachfolgenden Abschnitten hinzugefügt.

**TABELLE 5-4** Anpassungen von CPU-Einschätzungen für unterstützende Komponenten

| <b>Komponente</b>                      | <b>CPUs</b> | <b>Arbeitsspeicher</b>                      |
|--|-------------|---|
| Portal Server                          | 4           | 8 GB  |
| Communications Express                 | 2           | 4 GB  |
| Messaging Server (MTA, Eingang)        | 2           | 4 GB  |
| Messaging Server (MTA, Ausgang)        | 2           | 4 GB  |
| Messaging Server (MMP)                 | 2           | 4 GB  |
| Messaging Server (Nachrichtenspeicher) | 2           | 4 GB  |
| Access Manager                         | 2           | 4 GB  |
| Calendar Server                        | 2           | 4 GB  |
| Directory Server                       | 4           | 8 GB (aufgerundet von 3 CPUs/6 GB Speicher) |
| Web Server                             | 0           | 0   |



---

## Einschätzen von Prozessoranforderungen für sichere Transaktionen

Der sichere Transport von Daten beinhaltet die Verarbeitung von Transaktionen über ein sicheres Transportprotokoll wie Secure Sockets Layer (SSL) oder Transport Layer Security (TLS). Für Transaktionen, die im Rahmen eines sicheren Transports durchgeführt werden, ist normalerweise zusätzliche Rechnerkapazität erforderlich, um zunächst eine sichere Sitzung (die als Handshake bezeichnet wird) einzurichten und dann die übermittelten Daten zu ver- und entschlüsseln. Abhängig vom verwendeten Verschlüsselungsalgorithmus (z. B. des 40-Bit- oder des 128-Bit-Verschlüsselungsalgorithmus) ist möglicherweise sehr viel zusätzliche Rechnerkapazität erforderlich.

Damit sichere Transaktionen auf derselben Ebene wie nicht sichere Transaktionen ausgeführt werden können, muss zusätzliche Rechnerkapazität eingeplant werden. Je nachdem, welche Art von Transaktion ausgeführt wird und welche Sun Java™ Enterprise System-Dienste hierfür genutzt werden, ist für sichere Transaktionen u. U. das Vierfache an Rechnerkapazität erforderlich als bei nicht sicheren Transaktionen.

Um einzuschätzen, wie viel Verarbeitungsleistung für die Handhabung sicherer Transaktionen erforderlich ist, analysieren Sie Anwendungsfälle, um den Prozentsatz an Transaktionen zu ermitteln, für die der sichere Transport erforderlich ist. Wenn die Leistungsanforderungen für sichere Transaktionen mit denen für nicht sichere Transaktionen übereinstimmen, passen Sie die CPU-Einschätzungen dahin gehend an, dass die für sichere Transaktionen erforderliche Rechnerkapazität berücksichtigt wird.

In einigen Anwendungsszenarios ist der sichere Transport möglicherweise nur zu Authentifizierungszwecken erforderlich. Sobald ein Benutzer für das System authentifiziert wurde, sind keine zusätzlichen Sicherheitsmaßnahmen für den Datentransport erforderlich. In anderen Szenarios ist der sichere Transport möglicherweise für sämtliche Transaktionen erforderlich.

Wenn Sie beispielsweise den Produktkatalog einer E-Commerce-Site im Internet anzeigen, können sämtliche Transaktionen nicht sichere Transaktionen sein, bis der Kunde seine Wahl getroffen hat und einen Kauf tätigen möchte. In einigen Anwendungsszenarios, beispielsweise bei Bereitstellungen für Banken oder Brokerfirmen, müssen nahezu alle oder alle Transaktionen sicher sein; zudem muss für sichere und nicht sichere Transaktionen derselbe Leistungsstandard angewendet werden.

## CPU-Einschätzungen für sichere Transaktionen

In diesem Abschnitt wird die Beispielbereitstellung weiter erläutert, um aufzuzeigen, wie CPU-Anforderungen für einen theoretischen Anwendungsfall mit sicheren sowie nicht sichere Transaktionen berechnet werden können.

Um die CPU-Anforderungen für sichere Transaktionen einzuschätzen, führen Sie folgende Berechnungen durch:

1. Beginnen Sie mit einer Basiszahl für die CPU-Einschätzungen (wie im vorherigen Abschnitt, „[Beispiel für das Einschätzen von Prozessoranforderungen](#)“ auf Seite 75 erläutert).
2. Berechnen Sie den Prozentsatz der Transaktionen, für die der sichere Transport erforderlich ist, und berechnen Sie dann die CPU-Einschätzungen für die sicheren Transaktionen.
3. Berechnen Sie die verringerten CPU-Einschätzungen für nicht sichere Transaktionen.
4. Zählen Sie die sicheren und die nicht sicheren Einschätzungen zusammen, um die CPU-Gesamteinschätzungen zu ermitteln.
5. Runden Sie die CPU-Gesamteinschätzung auf eine gerade Zahl auf.

In „[CPU-Einschätzungen für sichere Transaktionen](#)“ auf Seite 82 finden Sie eine Beispielberechnung, die auf Anwendungsfällen und Anwendungsanalysen für Portal Server basiert, bei denen von Folgendem ausgegangen wird:

- Für sämtliche Anmeldungen ist die sichere Authentifizierung erforderlich.
- Sämtliche Anmeldungen machen 10% der Portal Server-Gesamtauslastung aus.
- Die Leistungsanforderung für sichere Transaktionen ist mit der Leistungsanforderung für nicht sichere Transaktionen identisch.

Um die zusätzliche Rechnerkapazität für die Handhabung sicherer Transaktionen zu berücksichtigen, wird die Anzahl der CPUs für die Verarbeitung dieser Transaktionen um den Faktor 4 erhöht. Wie bei anderen CPU-Werten im Beispiel ist dieser Faktor willkürlich gewählt und dient lediglich der Veranschaulichung.

**TABELLE 5-5** Ändern von CPU-Einschätzungen für sichere Transaktionen

| Schritt | Beschreibung   | Berechnung  | Ergebnis |
|---------|--|---|----------|
| 1       | Starten Sie mit einer Basiseinschätzung für sämtliche Portal Server-Transaktionen. | Die Basiseinschätzung aus „ <a href="#">Anwendungsfälle auf Spitzenauslastung hin prüfen</a> “ auf Seite 78 beträgt 4 CPUs. | -----    |

**TABELLE 5-5** Ändern von CPU-Einschätzungen für sichere Transaktionen (Fortsetzung)

| Schritt | Beschreibung   | Berechnung   | Ergebnis |
|---------|--|--|----------|
| 2       | Berechnen Sie die zusätzlichen CPU-Einschätzungen für sichere Transaktionen. Gehen Sie hierbei davon aus, dass für sichere Transaktionen im Vergleich zu nicht sicheren Transaktionen das 4fache an Prozessorkapazität erforderlich ist. | Für 10 Prozent der Basiseinschätzung ist der sichere Transport erforderlich:<br>$0.10 \times 4 \text{ CPUs} = 0.4 \text{ CPUs}$<br>Erhöhen Sie die CPU-Kapazität für sichere Transaktionen um den Faktor 4:<br>$4 \times 0.4 = 1.6 \text{ CPUs}$ | 1,6 CPUs |
| 3       | Berechnen Sie die verringerten CPU-Einschätzungen für nicht sichere Transaktionen.   | 90 Prozent der Basiseinschätzung sind nicht sicher:<br>$0.9 \times 4 \text{ CPUs} = 3.6 \text{ CPUs}$  | 3,6 CPUs |
| 4       | Berechnen Sie die angepassten CPU-Gesamteinschätzungen für sichere und nicht sichere Transaktionen.  | Sichere Einschätzung + nicht sichere Einschätzung = Gesamtwert:<br>$1.6 \text{ CPUs} + 3.6 \text{ CPUs} = 5.2 \text{ CPUs}$  | 5,2 CPUs |
| 5       | Runden Sie auf eine gerade Zahl auf.   | $5.2 \text{ CPUs} \implies 6 \text{ CPUs}$   | 6 CPUs   |

Von den Berechnungen für sichere Transaktionen in diesem Beispiel würden Sie die CPU-Gesamteinschätzungen in „CPU-Einschätzungen für sichere Transaktionen“ auf Seite 82 dahin gehend ändern, dass zwei zusätzliche CPUs und 4 GB Speicher hinzugefügt und so der Gesamtwert für Portal Server ermittelt wird.

| Komponente    | CPUs | Arbeitsspeicher |
|---------------|------|-----------------|
| Portal Server | 6    | 12 GB           |

## Spezielle Hardware zur Handhabung von SSL-Transaktionen

Spezielle Hardwaregeräte, beispielweise SSL-Beschleunigungskarten und weitere Vorrichtungen, können Rechnerkapazität für die Handhabung der Einrichtung sicherer Sitzungen und der Ver- und Entschlüsselung von Daten bereitstellen. Wenn Sie für SSL-Vorgänge spezielle Hardware einsetzen, wird ein Teil der Rechnerkapazität für SSL-Berechnungen genutzt, normalerweise für den so genannten Handshake-Vorgang, mit dem eine sichere Sitzung eingerichtet wird.

Diese Hardware erweist sich in Ihrer endgültigen Bereitstellungsarchitektur möglicherweise als vorteilhaft. Aufgrund der Spezialisierung der Hardware sollten Sie die Leistungsanforderungen für sichere Transaktionen zunächst im Hinblick auf die CPU-Kapazität einschätzen und erst dann die Vorteile der Verwendung spezieller Hardware zur Handhabung der zusätzlichen Last in Betracht ziehen.

Zu den Faktoren, die bei der Erwägung der Verwendung spezieller Hardware in Betracht gezogen werden müssen, zählt, ob die Anwendungsfälle die Hardware unterstützen (z. B. Anwendungsfälle, für die zahlreiche SSL-Handshake-Vorgänge erforderlich sind) sowie die erhöhte Komplexität, die diese Art von Hardware mit sich bringt. Diese Komplexität ergibt sich aus der Installation, der Konfiguration, dem Testen und Verwalten dieser Geräte.

---

## Ermitteln von Verfügbarkeitsstrategien

Ziehen Sie bei der Ausarbeitung einer Strategie für Verfügbarkeitsanforderungen die Interaktion zwischen Komponenten sowie die Anwendungsanalyse in Betracht, um zu ermitteln, welche Verfügbarkeitslösungen infrage kommen. Gehen Sie bei der Analyse jede Komponente einzeln durch und ermitteln Sie so die in Bezug auf Verfügbarkeit und Failover am besten geeignete Lösung.

Nachfolgend sind Beispiele für Angaben aufgeführt, die bei der Ermittlung von Verfügbarkeitsstrategien hilfreich sind:

- Wie viele Verfügbarkeits-Neunen sind angegeben?
- Wie sehen die Leistungsspezifikationen in Bezug auf Failover-Situationen aus (z. B. mindestens 50 % Leistung bei Failover)?
- Gehen aus der Anwendungsanalyse Zeiten mit Spitzenauslastung und andere Zeiten hervor?
- Welche geografischen Überlegungen gibt es?

In der von Ihnen ausgewählten Verfügbarkeitsstrategie müssen auch die Anforderungen hinsichtlich der Wartungseignung berücksichtigt werden, die unter „Konzeption für optimale Ressourcennutzung“ auf Seite 98 beschrieben sind. Komplexe Lösungen mit großem Verwaltungs- und Wartungsaufwand sollten vermieden werden.

## Verfügbarkeitsstrategien


Die Verfügbarkeitsstrategien für Java Enterprise System-Bereitstellungen beinhalten folgende Punkte:

- *Lastenausgleich*. Nutzt redundante Hard- und Softwarekomponenten zur verteilten Lastenverarbeitung. Von einem Lastenausgleichssystem werden sämtliche Anforderungen eines Dienstes an eine von mehreren symmetrischen Instanzen des Dienstes weitergeleitet. Wenn eine der Instanzen ausfällt, sind andere Instanzen verfügbar, die eine größere Last übernehmen.
- *Failover*. Umfasst die Verwaltung redundanter Hard- und Software zur Gewährleistung des ununterbrochenen Zugriffs von Diensten und der Sicherheit kritischer Daten beim Ausfall einer Komponente.  
Sun Cluster-Software beinhaltet eine Failover-Lösung für kritische Daten, die von Back-End-Komponenten verwaltet werden, beispielsweise der Nachrichtenspeicher für Messaging Server und Kalenderdaten für Calendar Server.
- *Replikation von Diensten*. Durch die Replikation von Diensten werden mehrere Zugriffsmöglichkeiten auf dieselben Daten bereitgestellt. Directory Server enthält zahlreiche Replikations- und Synchronisierungsstrategien für den LDAP-Verzeichniszugriff.

Die nachfolgenden Abschnitte enthalten Beispiele für Verfügbarkeitslösungen, die unterschiedliche Ebenen hinsichtlich Lastenausgleich, Failover und Replikation von Diensten bereitstellen.

## System mit nur einem Server

Platzieren Sie alle Rechnerressourcen für einen Dienst auf einem einzelnen Server. Wenn der Server ausfällt, schlägt der gesamte Dienst fehl.

|  |   |
|--|---|
| <p><b>Erforderliche Leistung: 10 CPUs</b></p>  <p>10 CPUs</p> | <p>Platzieren Sie alle CPUs, die die Leistungsanforderungen erfüllen auf einem einzelnen Server.</p> <p>Der Server ist ein Single Point of Failure.</p> |
|--|---|

**ABBILDUNG 5-2** System mit nur einem Server

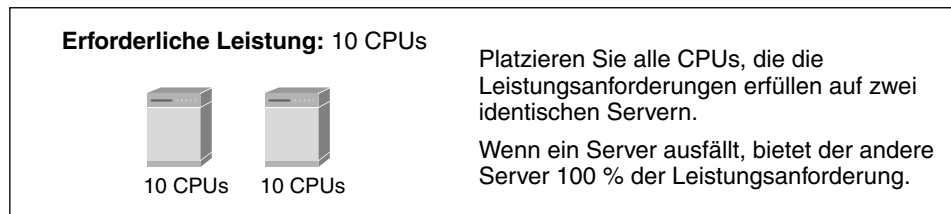
Sun bietet High-End-Server mit folgenden Vorteilen:

- Austausch und Neukonfiguration von Hardwarekomponenten im laufenden Betrieb
- Ausführung mehrerer Anwendungen in fehlerisolierten Domänen auf dem Server
- Aufrüstung hinsichtlich Kapazität und Leistungsgeschwindigkeit sowie E/A-Konfiguration ohne Neustart des Systems

Ein High-End-Server ist in der Regel teurer als ein vergleichbares System mit mehreren Servern. Ein einzelner Server hilft Kosten hinsichtlich Verwaltung, Überwachung und Hosting für Server in einem Datenzentrum zu sparen. Lastenausgleich, Failover und Entfernung von Single-Point-of-Failure gestalten sich in einem System mit mehreren Servern flexibler.

## Horizontal redundante Systeme

Es gibt mehrere Möglichkeiten, die Verfügbarkeit mit parallel redundanten Servern zu erhöhen, die sowohl Lastenausgleich als auch Failover bieten. In der nachfolgenden Abbildung sind zwei Replikationsserver dargestellt, die ein N+1-Failover-System zur Verfügung stellen. In einem N+1-System ist ein zusätzlicher Server vorhanden, der bei Ausfall eines Servers 100 %-ige Kapazität bereitstellt.

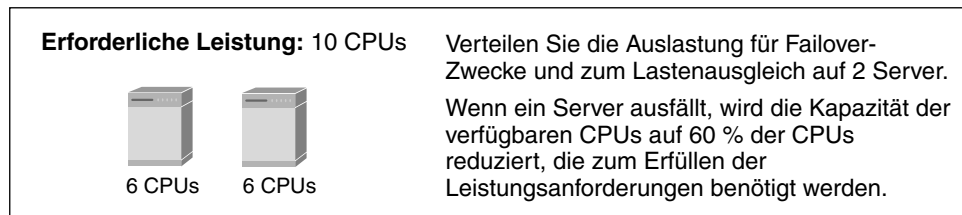


**ABBILDUNG 5-3** N+1-Failover-System mit zwei Servern

Die Rechnerkapazität der Server in der oben gezeigten Abbildung „[Horizontal redundante Systeme](#)“ auf Seite 86 ist identisch. Die Leistungsanforderungen werden von einem Server allein gehandhabt. Der andere Server stellt 100 % Leistung bereit, wenn er als Reserve in Anspruch genommen wird.

Der Vorteil eines N+1-Failover-Konzepts besteht darin, dass während eines Failovers 100 % Leistung bereitgestellt werden können. Zu den Nachteilen zählen die höheren Hardwarekosten, denen keine entsprechende Steigerung der Gesamtleistung gegenübersteht (da ein Server ausschließlich für den Einsatz in Failover-Situationen vorgesehen ist).

In der folgenden Abbildung ist ein System dargestellt, in dem Lastenausgleich plus Failover implementiert werden und so die Leistung zwischen zwei Servern aufgeteilt wird.

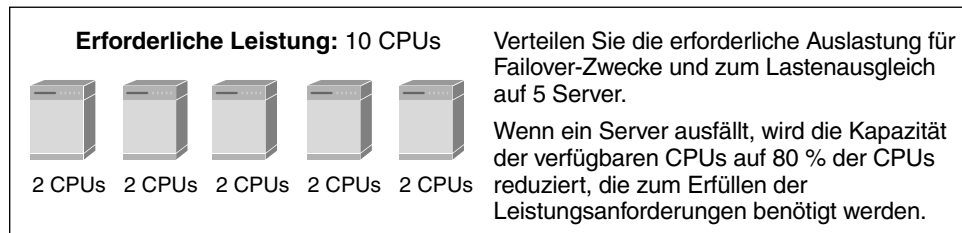


**ABBILDUNG 5-4** Lastenausgleich plus Failover zwischen zwei Servern

In dem System, das in „Horizontal redundante Systeme“ auf Seite 86 oben dargestellt ist, stehen beim Ausfall eines Servers sämtliche Dienste zur Verfügung, jedoch nur zu einem bestimmten Prozentsatz der vollen Kapazität. Der verbleibende Server stellt die Rechnerkapazität von 6 CPUs bereit, es werden also 60 % der 10-CPU-Anforderung erfüllt.

Ein Vorteil dieses Konzepts besteht in der zusätzlichen latenten Kapazität von 2 CPUs, wenn beide Server verfügbar sind.

Aus der folgenden Abbildung geht die Verteilung zwischen mehreren Servern in Bezug auf Leistung und Lastenausgleich hervor.



**ABBILDUNG 5-5** Verteilung der Last zwischen  $n$  Servern

Da das in der Abbildung „Horizontal redundante Systeme“ auf Seite 86 dargestellte Konzept 5 Server umfasst, stellen beim Ausfall eines Servers die verbleibenden Server die Rechnerkapazität von insgesamt 8 CPUs bereit; dies entspricht 80 % der 10-CPU-Leistungsanforderung. Wenn Sie dem Konzept einen zusätzlichen Server mit einer Kapazität von 2 CPUs hinzufügen, verfügen Sie über ein  $n+1$ -Konzept. Wenn ein Server ausfällt, wird die Leistungsanforderung von den verbleibenden Servern zu 100 % erfüllt.

Dieses Konzept bringt folgende Vorteile mit sich:

- Mehr Leistung bei Ausfall eines einzelnen Servers
- Gewährleistete Verfügbarkeit sogar beim Ausfall mehrerer Server
- Server können zu Wartungs- und Aktualisierungszwecken abwechselnd aus dem aktiven Betrieb genommen werden

- Mehrere Low-End-Server sind in der Regel kostengünstiger als ein einzelner High-End-Server

Durch zusätzliche Server können die Verwaltungs- und Wartungskosten jedoch deutlich ansteigen. Zudem müssen die Kosten für das Hosting der Server in einem Datenzentrum berücksichtigt werden. Ab einem bestimmten Punkt führt das Hinzufügen weiterer Server zu einer verringerten Rendite.

## Sun Cluster-Software

In Situationen, in denen ein hohes Maß an Verfügbarkeit erforderlich ist (z. B. bei vier oder fünf Neunen), sollten Sie Sun Cluster-Software als Bestandteil Ihres Verfügbarkeitskonzepts in Erwägung ziehen. Bei einem Cluster-System handelt es sich um die Kombination redundanter Server mit Speicher und anderen Netzwerkressourcen. Die Server in einem Cluster kommunizieren ständig miteinander. Wenn einer der Server in den Offline-Modus versetzt wird, wird er von den verbleibenden Geräten im Cluster isoliert und der Failover-Vorgang für jegliche Anwendungen oder Daten vom deaktivierten Knoten in einen anderen Knoten durchgeführt. Der Failover-Vorgang läuft relativ schnell und mit nur geringer Beeinträchtigung der Dienstbereitstellung für die Benutzer des Systems ab.

Für Sun Cluster-Software sind zusätzliche dedizierte Hardware und besondere Kenntnisse hinsichtlich Konfiguration, Verwaltung und Wartung erforderlich.

## Beispiele für Verfügbarkeitskonzepte

Dieser Abschnitt enthält zwei Beispiele für Verfügbarkeitsstrategien, die auf der identitätsbasierten Kommunikationslösung eines mittelständischen Unternehmens (1000 bis 5000 Mitarbeiter) basieren, die zuvor unter [„Identitätsbasiertes Kommunikationsbeispiel“](#) auf Seite 63 erläutert wurden. In der ersten Verfügbarkeitsstrategie ist der Lastenausgleich für Messaging Server dargestellt. In der zweiten Strategie wird eine Failover-Lösung erläutert, bei der Sun Cluster-Software zum Einsatz kommt.

## Beispiel des Lastenausgleichs für Messaging Server

In der folgenden Tabelle sind die Einschätzungen hinsichtlich CPU-Kapazität für die einzelnen logischen Messaging Server-Komponenten in der logischen Architektur aufgeführt. In dieser Tabelle wird die endgültige Einschätzung erneut verwendet, die im Abschnitt [„CPU-Einschätzungen aktualisieren“](#) auf Seite 80 berechnet wurden.



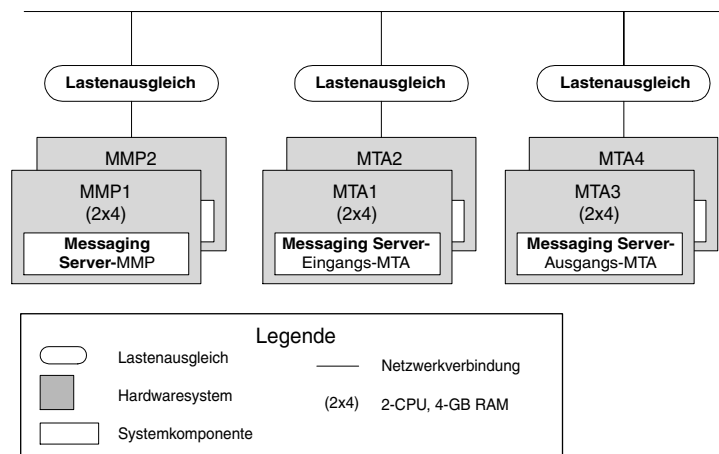
**TABELLE 5-6** Anpassungen von CPU-Einschätzungen für unterstützende Komponenten

| Komponente                             | CPUs | Arbeitsspeicher |
|--|------|-----------------|
| Messaging Server (MTA, Eingang)        | 2    | 4 GB            |
| Messaging Server (MTA, Ausgang)        | 2    | 4 GB            |
| Messaging Server (MMP)                 | 2    | 4 GB            |
| Messaging Server (Nachrichtenspeicher) | 2    | 4 GB            |

Gehen Sie bei diesem Beispiel davon aus, dass in der Phase der technischen Anforderungen folgende Dienstqualitätsanforderungen festgelegt wurden:

- **Verfügbarkeit.** Die Gesamtverfügbarkeit des Systems sollte 99,99 % betragen (geplante Ausfallzeiten sind hier nicht berücksichtigt). Der Ausfall eines einzelnen Computersystems sollte nicht zu einem Dienstaussfall führen.
- **Skalierbarkeit.** Kein Server sollte eine Auslastung von mehr als 80 % der täglichen Spitzenauslastung erreichen und im System muss das langfristige Wachstum von 10 % pro Jahr berücksichtigt werden.

Um die Verfügbarkeitsanforderung zu erfüllen, müssen für jede Messaging Server-Komponente zwei Instanzen bereitgestellt werden (jeweils eine auf separaten Servern). Wenn ein Server einer Komponente ausfällt, wird der Dienst vom anderen Server bereitgestellt. In der folgenden Abbildung wird das Netzwerkdiagramm für diese Verfügbarkeitsstrategie dargestellt.



In der obigen Darstellung hat sich die Anzahl der CPUs im Vergleich zur ursprünglichen Einschätzung verdoppelt. Für die Verdoppelung der CPUs gibt es folgende Gründe:

- Wenn ein Server ausfällt, stellt der verbleibende Server die CPU-Kapazität zur Handhabung der Last bereit.
- Hinsichtlich der Skalierbarkeitsanforderung, wonach kein einzelner Server zu mehr als 80 % der Spitzenauslastung beansprucht werden soll, wird dieser Sicherheitsspielraum durch die zusätzliche CPU-Kapazität bereitgestellt.
- Hinsichtlich der Skalierbarkeitsanforderung, wonach ein Lastanstieg von 10 % pro Jahr berücksichtigt werden soll, wird durch die zusätzliche CPU-Kapazität latente Kapazität verfügbar gemacht, die die wachsende Last abdeckt, bis zusätzliche Skalierung erforderlich wird.

## Failover-Beispiel mit Sun Cluster-Software

In der folgenden Abbildung wird ein Beispiel einer Failover-Strategie für das Calendar Server-Back-End und den Messaging Server-Datenspeicher dargestellt. Das Calendar Server-Back-End und der Nachrichtenspeicher werden auf separaten Servern repliziert und unter Verwendung von Sun Cluster-Software für das Failover konfiguriert. Die Anzahl an CPUs und der entsprechende Arbeitsspeicher werden in Sun Cluster auf jedem Server repliziert.

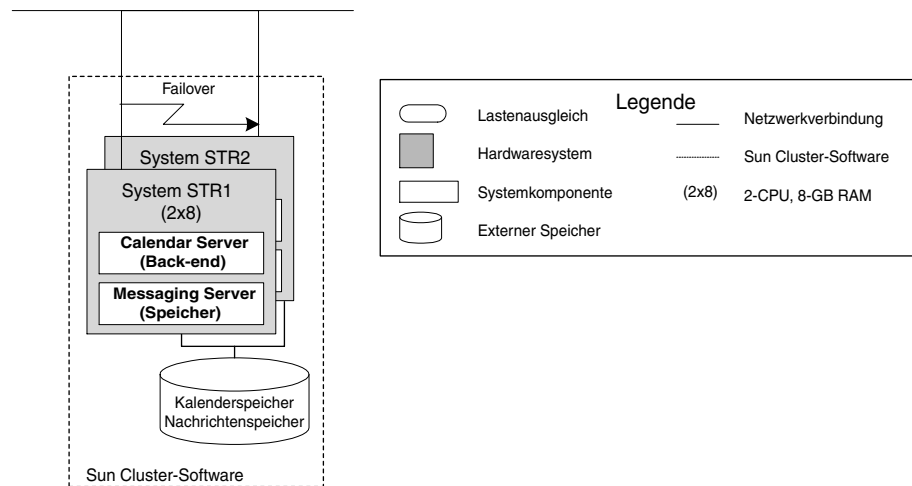


ABBILDUNG 5-6 Failover-Konzept mit Sun Cluster-Software

## Beispiel für Replikation von Verzeichnisdiensten

Verzeichnisdienste können repliziert werden, um Transaktionen auf unterschiedliche Server aufzuteilen und auf diese Weise die Hochverfügbarkeit zu gewährleisten. Directory Server bietet u. a. folgende Strategien für die Replikation von Diensten:

- **Mehrere Datenbanken.** Speichert unterschiedliche Bestandteile eines Verzeichnisbaums in separaten Datenbanken.
- **Verkettung und Referrals.** Nimmt die Verknüpfung verteilter Daten in einen einzelnen Verzeichnisbaum vor.
- **Einzelmaster-Replikation.** Stellt eine zentrale Quelle für die Master-Datenbank bereit, die im Anschluss auf Verbraucher-Replikate aufgeteilt wird.
- **Multimaster-Replikation.** Verteilt die Master-Datenbank auf mehrere Server. Von jedem dieser Master wird die zugehörige Datenbank auf Verbraucher-Replikate verteilt.

Verfügbarkeitsstrategien für Directory Server sind ein komplexes Thema, das den Rahmen dieses Handbuchs sprengen würde. In den folgenden Abschnitten, „[Einzelmaster-Replikation](#)“ auf Seite 91 und „[Multimaster-Replikation](#)“ auf Seite 92 werden die grundlegenden Replikationsstrategien erläutert. Ausführliche Informationen finden Sie in Kapitel 12, „[Designing a Highly Available Deployment](#),“ im *Sun Java System Directory Server 5 2005Q1 Deployment Planning Guide*.

## Einzelmaster-Replikation

In der folgenden Abbildung ist eine Einzelmaster-Replikationsstrategie dargestellt, anhand der grundlegende Replikationskonzepte veranschaulicht werden.

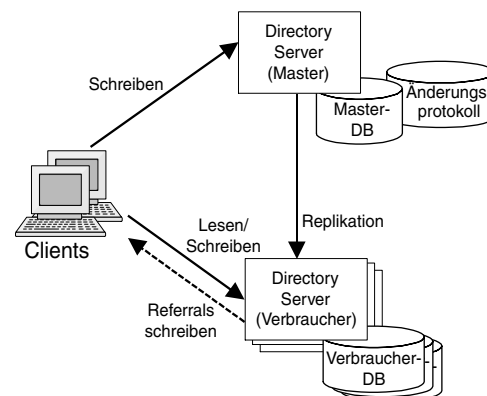


ABBILDUNG 5-7 Beispiel für eine Einzelmaster-Replikation

In einer Einzelmaster-Replikation verwaltet eine Instanz von Directory Server die Master-Verzeichnisdatenbank und protokolliert sämtliche Änderungen. Die Master-Datenbank wird auf einer beliebigen Anzahl an Verbraucher-Datenbanken

repliziert. Die Verbraucher-Instanzen von Directory Server werden für Lese- und Suchvorgänge optimiert. Sämtliche Schreibvorgänge, die von einem Verbraucher empfangen werden, werden an den Master zurückverwiesen. Die Verbraucher-Datenbanken werden in regelmäßigen Abständen vom Master aktualisiert.

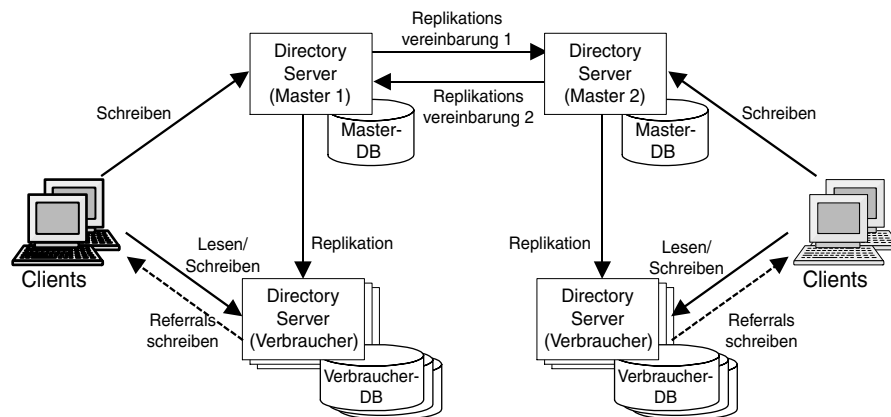
Zu den Vorteilen einer Einzelmaster-Replikation zählen:

- Einzelne für datenbankbezogene Lese- und Schreibvorgänge optimierte Directory Server-Instanz
- Eine beliebige Anzahl von für Lese- und Suchvorgänge optimierten Verbraucher-Instanzen von Directory Server
- Horizontale Skalierbarkeit für Verbraucher-Instanzen von Directory Server

## Multimaster-Replikation

In der folgenden Abbildung ist eine Multimaster-Replikationsstrategie dargestellt, die zur globalen Verteilung des Verzeichniszugriffs verwendet werden kann.

In einer Multimaster-Replikation wird die Master-Verzeichnisdatenbank von einer oder mehreren Instanzen von Directory Server verwaltet. Jeder Master weist eine Replikationsvereinbarung auf, aus der Vorgehensweisen für die Synchronisierung der Master-Datenbanken ersichtlich sind. Jeder Master wird auf eine beliebige Anzahl an Verbraucher-Datenbanken repliziert. Wie bei der Einzelmaster-Replikation werden die Verbraucher-Instanzen von Directory Server für den Lese- und Suchzugriff optimiert. Sämtliche Schreibvorgänge, die von einem Verbraucher empfangen werden, werden an den Master zurückverwiesen. Die Verbraucher-Datenbanken werden in regelmäßigen Abständen vom Master aktualisiert.



ABILDUNG 5-8 Beispiel für eine Multimaster-Replikation

Die Multimaster-Replikationsstrategie bietet alle Vorteile einer Einzelmaster-Replikation und zusätzlich eine Verfügbarkeitsstrategie, durch die der Lastenausgleich für Aktualisierungen des Masters bereitgestellt werden kann. Sie können auch eine Verfügbarkeitsstrategie implementieren, die die lokale Steuerung von Verzeichniseingängen ermöglicht, ein wichtiger Aspekt für Unternehmen mit global verteilten Datenzentren.

---

## Festlegen von Strategien hinsichtlich der Skalierbarkeit

Skalierbarkeit ist die Fähigkeit, Ihrem System mehr Kapazität zur Verfügung zu stellen; normalerweise durch das Hinzufügen von Systemressourcen ohne Änderung der Bereitstellungsarchitektur. Bei der Anforderungsanalyse werden üblicherweise Prognosen hinsichtlich des erwarteten Zuwachses eines Systems abgegeben. Als Grundlage dienen hier die Geschäftsanforderungen und die nachfolgende Anwendungsanalyse. Diese Prognosen der Anzahl der Benutzer eines Systems und der Kapazität des Systems zur Erfüllung der jeweiligen Anforderungen sind häufig Einschätzungen, die stark von den tatsächlichen Werten für das bereitgestellte System abweichen. Ihr Konzept sollte so flexibel sein, dass Abweichungen in Ihren Prognosen kein Problem darstellen.

Ein skalierbares Konzept stellt ausreichend latente Kapazität zur Handhabung einer größeren Last bereit, bis ein System mit weiteren Ressourcen aktualisiert werden kann. Skalierbare Konzepte können ohne weiteres so skaliert werden, dass die Handhabung einer größeren Last ohne Neukonzeption des Systems ermöglicht wird.

## Latente Kapazität

Die latente Kapazität ist ein Aspekt der Skalierbarkeit, bei dem zusätzliche Leistung und Verfügbarkeitsressourcen für das System bereitgestellt werden können, das daraufhin in der Lage ist, ungewöhnliche Spitzenauslastung zu handhaben. Sie können die Verwendung der latenten Kapazität in einem bereitgestellten System überwachen, um zu ermitteln, wie das System durch das Hinzufügen von Ressourcen skaliert werden kann. Die latente Kapazität stellt eine der Möglichkeiten dar, mit der Sicherheitsaspekte in Ihr Konzept integriert werden können.

Die Analyse von Anwendungsfällen kann bei der Ermittlung von Szenarios hilfreich sein, in denen es zu ungewöhnlicher Spitzenauslastung kommen kann. Nutzen Sie diese Analyse ungewöhnlicher Spitzenauslastung und einen Faktor zur Abdeckung unerwarteten Zuwachses, um latente Kapazität verfügbar zu machen, durch die die Sicherheit Ihres Systems erhöht wird.

Ihr System sollte in der Lage sein, die geplante Kapazität für einen angemessenen Zeitraum zu handhaben, normalerweise für die ersten 6 bis 12 Monate des Betriebs. Wartungszyklen können verwendet werden, um Ressourcen hinzuzufügen oder die Kapazität nach Bedarf zu erhöhen. Im Idealfall sollten Sie in der Lage sein, Aktualisierungen des Systems in regelmäßigen Abständen zu planen, die Vorhersage der erforderlichen Kapazitätserhöhungen erweist sich jedoch in vielen Fällen als schwierig. Verlassen Sie sich bei der Ermittlung des richtigen Zeitpunkts für die Aktualisierung eines Systems auf die genaue Überwachung Ihrer Ressourcen sowie auf Geschäftsprognosen.

Wenn Sie die phasenweise Implementierung Ihrer Lösung beabsichtigen, können Sie die Erhöhung der Kapazität so planen, dass sie mit anderen Verbesserungen für jede Phase zusammenfällt.

## Skalierbarkeitsbeispiel

Anhand des Beispiels in diesem Abschnitt wird die horizontale und vertikale Skalierung einer Lösung veranschaulicht, mit der Messaging Server implementiert wird. Für die horizontale Skalierung fügen Sie einem Server für die Handhabung steigender Last zusätzliche CPUs hinzu. Für die vertikale Skalierung wird die steigende Last durch das Hinzufügen weiterer Server gehandhabt, auf die die Last aufgeteilt wird.

Bei diesem Beispiel wird von einem Benutzerstamm mit 50.000 Benutzern ausgegangen, der durch zwei Nachrichtenspeicherinstanzen unterstützt wird, die für den Lastenausgleich aufgeteilt werden. Jeder Server verfügt über zwei CPUs, insgesamt stehen also vier CPUs zur Verfügung. Aus der nachfolgenden Abbildung geht hervor, wie dieses System zur Handhabung der ansteigenden Last für 250.000 Benutzer und 2.000.000 Benutzer skaliert werden kann.

---

**Hinweis** – In der Abbildung „Skalierbarkeitsbeispiel“ auf Seite 94 wird der Unterschied zwischen vertikaler und horizontaler Skalierung erläutert. Es werden keine zusätzlichen Faktoren veranschaulicht, die bei der Skalierung berücksichtigt werden müssen, beispielsweise Lastenausgleich, Failover sowie Änderungen in Anwendungsmustern.

---

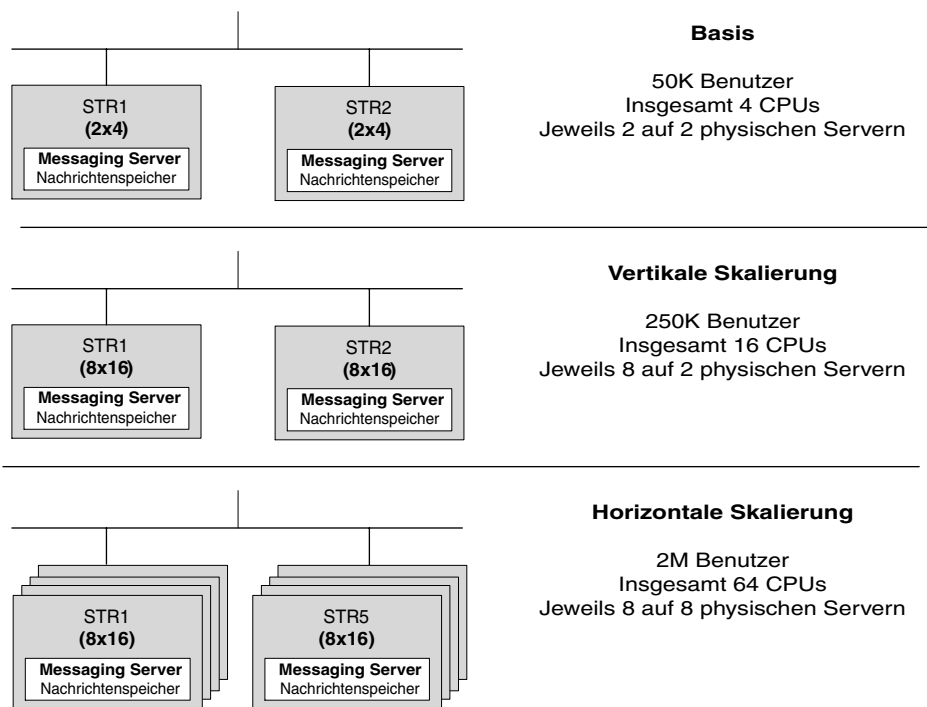


ABBILDUNG 5-9 Beispiele für horizontale und vertikale Skalierung

## Identifizieren von Leistungsengpässen

Einer der Schlüssel für ein erfolgreiches Bereitstellungskonzept ist das Identifizieren potenzieller Leistungsengpässe und das Ausarbeiten einer Strategie zu deren Vermeidung. Zu einem Leistungsengpass kommt es, wenn die Datenzugriffsrate vorgegebene Systemanforderungen nicht erfüllen kann.

Engpässe können anhand unterschiedlicher Hardwareklassen in Kategorien eingeteilt werden, wie in der folgenden Tabelle mit Datenzugriffspunkten innerhalb eines Systems angegeben. In dieser Tabelle werden zudem mögliche Lösungsvorschläge für Engpässe in den einzelnen Hardwareklassen unterbreitet.

TABELLE 5-7 Datenzugriffspunkte

| Hardwareklasse                           | Relative Zugriffsgeschwindigkeit   | Lösungsvorschläge zur Leistungssteigerung   |
|--|--|---|
| Prozessor                                | Nanosekunden   | Vertikale Skalierung: Mehr Verarbeitungsleistung bereitstellen, Prozessorcache vergrößern<br><br>Horizontale Skalierung: Parallele Verarbeitungsleistung für den Lastenausgleich bereitstellen  |
| Systemspeicher (RAM)                     | Mikrosekunden  | Systemspeicher bestimmten Aufgaben zuordnen<br><br>Vertikale Skalierung: Zusätzlichen Speicher bereitstellen<br><br>Horizontale Skalierung: Zusätzliche Instanzen für Parallelverarbeitung und Lastenausgleich erstellen  |
| Lese- und Schreibvorgänge auf Festplatte | Millisekunden  | Festplattenzugriff mithilfe von Festplattenarrays (Redundant Array of Independent Disks, RAID) optimieren<br><br>Festplattenzugriff bestimmten Funktionen zuordnen, beispielsweise dem reinen Lese- oder Schreibzugriff<br><br>Daten, auf die häufig zugegriffen wird, im Systempeicher zwischenspeichern |
| Netzwerk-schnittstelle                   | Variiert in Abhängigkeit von Bandbreite und Zugriffsgeschwindigkeit von Knoten im Netzwerk | Bandbreite erhöhen<br><br>Beschleunigungshardware für den Transport sicherer Daten bereitstellen<br><br>Leistung der Knoten im Netzwerk erhöhen, um höhere Datenverfügbarkeit zu gewährleisten  |

**Hinweis** – Unter „Identifizieren von Leistungsengpässen“ auf Seite 95 sind Hardwareklassen nach relativer Zugriffsgeschwindigkeit aufgeführt, hieraus lässt sich schließen, dass langsame Zugriffspunkte, beispielsweise Festplatten, mit größerer Wahrscheinlichkeit zu Engpässen führen. Prozessoren, die nicht genügend Leistung zur Handhabung einer großen Last aufweisen, können jedoch auch häufig die Ursache von Engpässen sein.



Am Anfang eines Bereitstellungskonzepts stehen in der Regel Basiseinschätzungen der Verarbeitungsleistung sämtlicher Komponenten in der Bereitstellung sowie deren Abhängigkeiten. Im Anschluss wird ermittelt, wie Engpässe in Bezug auf Systemspeicher und Festplattenzugriff vermieden werden können. Abschließend wird die Prüfung der Netzwerkschnittstelle auf potenzielle Engpässe sowie die Ausarbeitung von Strategien, mit denen diese Engpässe beseitigt werden können, durchgeführt.

## Optimieren des Festplattenzugriffs

Eine wichtige Komponente des Bereitstellungskonzepts ist die Geschwindigkeit des Festplattenzugriffs auf häufig genutzte Datengruppen, beispielsweise LDAP-Verzeichnisse. Der Festplattenzugriff ist die langsamste Art des Zugriffs auf Daten und häufig die Ursache für leistungsbezogene Engpässe.

Eine Möglichkeit zur Optimierung des Festplattenzugriffs besteht darin, Lese- und Schreibvorgänge voneinander zu trennen. Schreibvorgänge sind nicht nur kostspieliger als Lesevorgänge, Lesevorgänge (Suchvorgänge für LDAP-Verzeichnisse) treten normalerweise deutlich häufiger auf als Schreibvorgänge (Aktualisierungen von Daten in LDAP-Verzeichnissen).

Eine weitere Möglichkeit zur Optimierung des Festplattenzugriffs besteht darin, Datenträger für unterschiedliche E/A-Vorgänge vorzusehen. Sie können beispielsweise den separaten Festplattenzugriff für Protokollierungsvorgänge (z. B. die Transaktions- und Ereignisprotokolle) von Directory Server und für LDAP-Schreib- und Lesevorgänge ermöglichen.

Außerdem sollten Sie die Implementierung einer oder mehrerer Instanzen von Directory Server, die für Lese- und Schreibvorgänge vorgesehen sind, sowie die Verteilung replizierter Instanzen auf lokale Server für den Lese- und Suchzugriff in Erwägung ziehen. Zudem stehen Optionen für Verkettung und Verknüpfung zur Verfügung, mit denen sich der Zugriff auf Verzeichnisdienste optimieren lässt.

In Kapitel 6, „Defining System Characteristics,“ im *Sun Java System Directory Server 5 2005Q1 Deployment Planning Guide* werden verschiedene für die Planung des Festplattenzugriffs zu berücksichtigende Faktoren erläutert. Folgende Themen werden in diesem Kapitel behandelt:

- **Minimum memory and disk space requirements** Bietet Einschätzungen dazu, wie viel Festplatten- und Speicherplatz für Verzeichnisse unterschiedlicher Größe benötigt wird.
- **Sizing physical memory for cache access** Enthält Anleitungen zur Einschätzung der Cachegröße gemäß der geplanten Nutzung von Directory Server sowie zur Planung der Gesamtspeichernutzung.
- **Sizing disk subsystems** Enthält Informationen zur Planung der Festplattengröße gemäß den Verzeichnissuffixe und zu Directory Server-Faktoren, die sich auf die Datenträgenutzung auswirken. Außerdem wird erläutert, wie Dateien

datenträgerübergreifend verteilt werden und welche Alternativen es im Hinblick auf Festplattenarrays gibt.

---

## Konzeption für optimale Ressourcennutzung

In einem Bereitstellungskonzept müssen nicht nur die Ressourcen eingeschätzt werden, die für die Erfüllung von QoS-Anforderungen erforderlich sind. In der Konzeptionsphase müssen vielmehr alle verfügbaren Optionen analysiert und die beste Lösung ausgewählt werden, bei der bei geringstmöglichen Kosten die Erfüllung der QoS-Anforderungen gewährleistet werden kann. Sie müssen den Tradeoff sämtlicher Konzeptentscheidungen analysieren, um sicherzustellen, dass der in einem Bereich erzielte Vorteil nicht von unverhältnismäßigen Kosten in einem anderen aufgehoben wird.

So wird beispielsweise durch die horizontale Skalierung aus Verfügbarkeitsgründen möglicherweise die Gesamtverfügbarkeit erhöht, es ist jedoch auch ein höherer Kostenaufwand im Hinblick auf Wartung und Service vonnöten. Durch die vertikale Skalierung aus Leistungsgründen wird möglicherweise die Rechnerkapazität auf kostengünstige Weise erhöht, es kann jedoch sein, dass die zusätzliche Kapazität von einigen Diensten nicht effizient genutzt wird.

Überprüfen Sie vor Fertigstellung Ihrer Konzeptstrategie Ihre Entscheidungen nochmals, um sicherzustellen, dass ein Gleichgewicht zwischen der Nutzung von Ressourcen und dem Gesamtvorteil für die vorgeschlagene Lösung erzielt wurde. Bei dieser Analyse wird üblicherweise geprüft, wie sich Systemqualitäten in einem Bereich auf andere Systemqualitäten auswirken. In der folgenden Tabelle sind einige Systemqualitäten und die zugehörigen Überlegungen im Hinblick auf die Ressourcenverwaltung aufgeführt.

**TABELLE 5–8** Überlegungen hinsichtlich Ressourcenverwaltung

| Systemqualität | Beschreibung  |
|----------------|---|
| Leistung       | Können die Dienste bei Leistungslösungen, bei denen die einzelnen Server eine hohe Zahl an CPUs aufweisen, mit Schwerpunkt die Rechnerkapazität effizient nutzen? (Bei einigen Diensten gibt es beispielsweise eine Obergrenze hinsichtlich der Anzahl an CPUs, die effizient genutzt werden können.) |

**TABELLE 5–8** Überlegungen hinsichtlich Ressourcenverwaltung (Fortsetzung)

| Systemqualität    | Beschreibung  |
|-------------------|---|
| Latente Kapazität | <p>Kann mit Ihrer Strategie eine Last gehandhabt werden, die die Leistungseinschätzungen übersteigt?</p> <p>Wird die übermäßige Last durch die vertikale Skalierung auf Server, den Lastenausgleich auf andere Server oder durch beides gehandhabt?</p> <p>Reicht die latente Kapazität zur Handhabung ungewöhnlicher Spitzenlast aus, bis der nächste Meilenstein der Bereitstellungsskalierung erreicht wird?</p> |
| Sicherheit        | <p>Haben Sie den leistungsbezogenen Mehraufwand, der für die Handhabung sicherer Transaktionen erforderlich ist, ausreichend berücksichtigt?</p>  |
| Verfügbarkeit     | <p>Haben Sie in Bezug auf horizontal redundante Lösungen die langfristigen Wartungskosten ausreichend berücksichtigt?</p> <p>Haben Sie die geplanten Ausfallzeiten berücksichtigt, die für die Wartung des Systems erforderlich sind?</p> <p>Wurde ein Kostengleichgewicht zwischen High-End- und Low-End-Servern erzielt?</p>  |
| Skalierbarkeit    | <p>Haben Sie in Ihrer Einschätzung Meilensteine für die Bereitstellungsskalierung berücksichtigt?</p> <p>Verfügen Sie über eine Strategie, mit der ausreichend latente Kapazität zur Handhabung von prognostiziertem Lastanstieg bereitgestellt wird, bis die Meilensteine für die Skalierung der Bereitstellung erreicht werden?</p>   |
| Wartungseignung   | <p>Haben Sie die Kosten für Verwaltung, Überwachung und Wartung in Ihrem Verfügbarkeitskonzept berücksichtigt?</p> <p>Haben Sie Lösungen für die delegierte Verwaltung (bei denen Endbenutzern die Ausführung einiger Verwaltungsaufgaben ermöglicht wird) in Erwägung gezogen, um Verwaltungskosten zu senken?</p>   |

---

## Verwalten von Risiken

Beim Großteil der Informationen, auf denen sich ein Bereitstellungskonzept stützt, beispielsweise die Dienstqualitätsanforderungen und die Anwendungsanalyse, handelt es sich nicht um empirische Daten, sondern um Daten, die auf Einschätzungen und Prognosen basieren, die von Geschäftsanalysen abgeleitet wurden. Diese Prognosen können aus zahlreichen Gründen falsch sein, u. a. durch unvorhergesehene Umstände im Geschäftsklima, fehlerhafte Methoden der

Datenzusammenstellung oder einfach aus Fehlern, die Benutzern unterlaufen. Sehen Sie sich vor der Fertigstellung eines Bereitstellungskonzepts nochmals die Analysen an, auf denen das Konzept basiert, um sicherzustellen, dass in Ihrem Konzept etwaige realistische Abweichungen von Einschätzungen oder Prognosen berücksichtigt wurden.

Wenn beispielsweise in der Anwendungsanalyse die tatsächliche Nutzung des Systems zu niedrig eingeschätzt wurde, laufen Sie Gefahr, ein System zu konfigurieren, das dem anfallenden Datenverkehr nicht gewachsen ist. Ein Konzept, das den Leistungsvorgaben nicht gerecht wird, wird mit Sicherheit als Misserfolg betrachtet.

Wenn auf der anderen Seite ein System konfiguriert wird, das um einiges leistungsfähiger ist als erforderlich, werden Ressourcen abgezogen, die anderswo sinnvoll eingesetzt werden könnten. Der Schlüssel liegt darin, einen Sicherheitsspielraum einzuplanen, der die Anforderungen übertrifft, gleichzeitig aber den verschwenderischen und folglich nicht effizienten Einsatz von Ressourcen unterbindet.

Der verschwenderische Einsatz von Ressourcen ist ein Fehler des Konzepts, da die nicht genügend beanspruchten Ressourcen an anderer Stelle hätten eingesetzt werden können. Zudem könnten verschwenderische Lösungen von den Interessengruppen als Nichterfüllung von in gutem Glauben abgeschlossenen Verträgen interpretiert werden.

---

## Beispiel für eine Bereitstellungsarchitektur

In der folgenden Abbildung ist eine fertig gestellte Bereitstellungsarchitektur für die Beispielbereitstellung dargestellt, die weiter oben in diesem White Paper eingeführt wurde. Diese Abbildung dient als Anhaltspunkt für die Präsentation einer Bereitstellungsarchitektur.



---

**Achtung** – Die in der folgenden Abbildung dargestellte Bereitstellungsarchitektur dient lediglich der Veranschaulichung. Es handelt sich hierbei nicht um eine Bereitstellung, die tatsächlich konzipiert, erstellt oder getestet wurde, und sollte folglich bei der Bereitstellungsplanung nicht als Anhaltspunkt herangezogen werden.

---

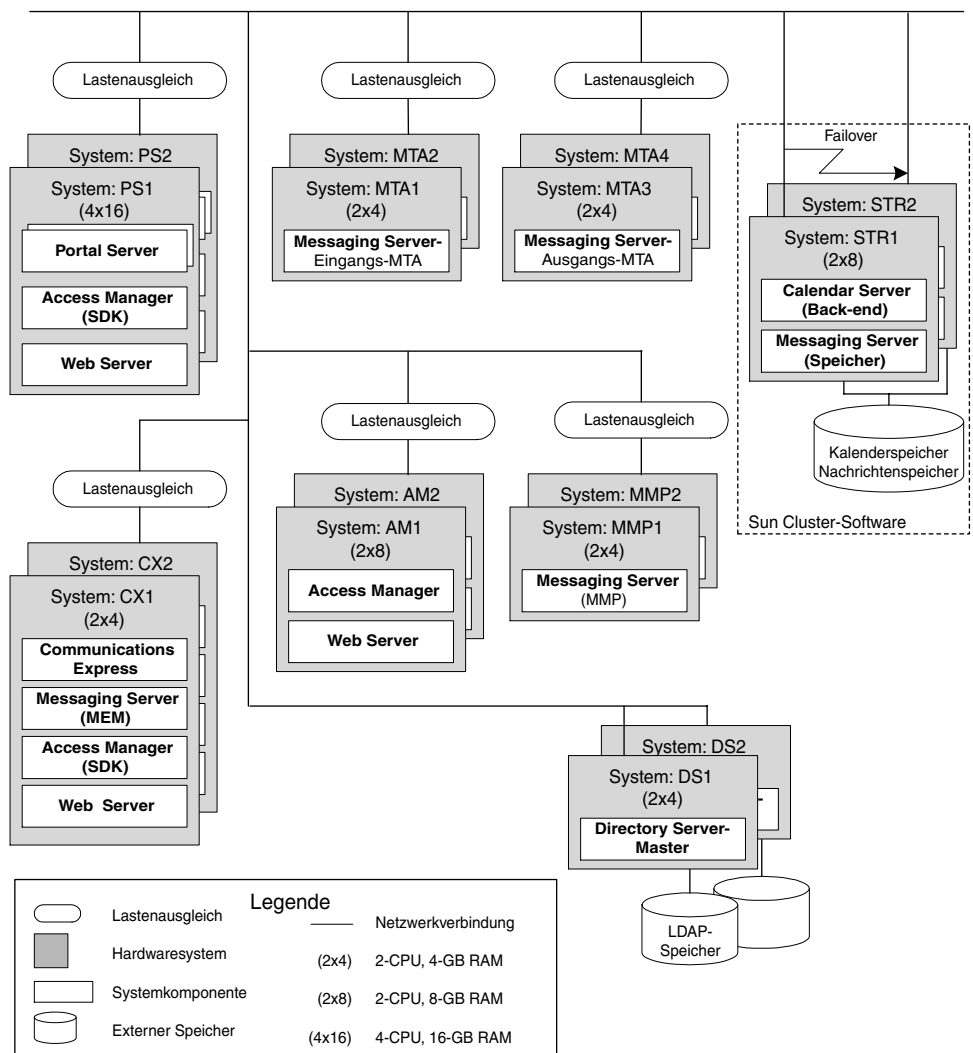


ABBILDUNG 5–10 Beispiel für eine Bereitstellungsarchitektur



## Implementierung eines Bereitstellungskonzepts

---

In der Implementierungsphase des Lebenszyklus einer Lösung arbeiten Sie mit Spezifikationen und Plänen, die Sie in der Phase des Bereitstellungskonzepts erstellt haben, um die Bereitstellungsarchitektur zu entwickeln und zu testen und die Bereitstellung schließlich in Produktion zu geben. Eine detaillierte Beschreibung der Implementierung an sich würde den Rahmen dieses Handbuchs sprengen, das vorliegende Kapitel bietet jedoch einen guten Überblick über diese Phase.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Informationen zur Implementierung des Bereitstellungskonzepts“ auf Seite 103
- „Installation und Konfiguration der Software“ auf Seite 104
- „Entwickeln von Pilotbereitstellungen und Prototypen“ auf Seite 105
- „Testen von Pilot- und Prototypbereitstellungen“ auf Seite 106
- „Einsetzen einer Produktionsbereitstellung“ auf Seite 107

---

## Informationen zur Implementierung des Bereitstellungskonzepts

Nachdem die Bereitstellungsarchitektur genehmigt wurde und die Implementierungsspezifikationen und -pläne abgeschlossen wurden, beginnt die Implementierungsphase des Lebenszyklus der Lösung. Die Implementierung ist eine komplexe Reihe von Vorgängen und Vorgehensweisen, die sorgfältig geplant werden muss, damit sie erfolgreich durchgeführt werden kann. Die Implementierung umfasst folgende Aufgaben:

- Einrichtung der Netzwerk- und Hardwareinfrastruktur
- Installation und Konfiguration der Software gemäß einem Installationsplan
- Migration von Daten aus vorhandenen Anwendungen in die aktuelle Lösung

- Implementierung eines Benutzerverwaltungsplans
- Entwurf und Bereitstellung von Pilotbereitstellungen oder Prototypen in einer Testumgebung gemäß einem Testplan
- Entwurf und Ausführung von Funktionstests und Belastungstests gemäß einem Testplan
- Einsetzen der Lösung aus Testumgebung in Produktionsumgebung gemäß einem Roll-out-Plan
- Schulung der Administratoren und Benutzer der Bereitstellung gemäß einem Schulungsplan

Die Details für die Implementierung würden den Rahmen dieses Handbuchs sprengen. Die folgenden Abschnitte vermitteln jedoch einen Überblick über einige dieser Aufgaben.

---

## Installation und Konfiguration der Software

Die Installation und Konfiguration von Sun Java™ Enterprise System für eine verteilte Unternehmensanwendung erfordert die Planung und Koordination vieler Aufgaben und Verfahren. In der Bereitstellungskonzeptphase erstellen Sie einen Installationsplan anhand der Bereitstellungsarchitektur auf hoher Ebene, der die für die Installation der Java Enterprise System-Software erforderlichen Installations- und Konfigurationsinformationen enthält.

Hauptpunkte dieses Installationsplans:

- Ermittlung der Installationsreihenfolge und des Installationstyps
- Prüfung der Hosts in Bezug auf zuvor installierte Software und Installationsbereitschaft
- Zusammenstellung von Konfigurationsinformationen für jede zu installierende Java Enterprise System-Komponente

Das Handbuch *Java ES System 2005Q4 Handbuch zur Installationsplanung* enthält Informationen dazu, wie man Informationen für einen Installationsplan zusammenstellt. Das Handbuch *Java ES System 2005Q4 Installation-Verweis* enthält detaillierte Konfigurationsinformationen und Arbeitsblätter, mit deren Hilfe Sie diese Informationen dokumentieren können. Das Handbuch *Sun Java Enterprise System 2005Q4 Installationshandbuch für UNIX* Anleitungen für häufige Installationsszenarien, die mehrere Java Enterprise System-Komponenten umfassen. Weitere Informationen finden Sie in Kapitel 1, „Vorbereitung der Installation“ in *Sun Java Enterprise System 2005Q4 Installationshandbuch für UNIX*.



---

## Entwickeln von Pilotbereitstellungen und Prototypen

Java Enterprise System-Bereitstellungen lassen sich normalerweise in zwei Kategorien unterteilen. Diejenigen, die in erster Linie auf Diensten basieren, die mit Java Enterprise System zur Verfügung gestellt werden, und diejenigen, die eine erhebliche Anzahl an benutzerdefinierten Diensten benötigen, die in die Java Enterprise System-Dienste integriert sind. Der erste Bereitstellungstyp kann als eine 80:20-Bereitstellung verstanden werden (80 % der Dienste werden von Java Enterprise System zur Verfügung gestellt) und der zweite entsprechend als eine 20:80-Bereitstellung.

Im Fall von 80:20-Bereitstellungen entwickeln Sie normalerweise in der Implementierungsphase eine Pilotbereitstellung zu Testzwecken. Da 80:20-Bereitstellungen voll entwickelte Java Enterprise System-Dienste verwenden, die gebrauchsfertige Funktionen bieten, gehen Pilotbereitstellungen relativ schnell von der Entwicklung, der Testphase und den damit verbundenen Änderungsschritten in Produktionsbereitstellungen über. Eine Pilotbereitstellung dient zum Überprüfen der Funktionalität einer Lösung, gibt aber auch Aufschluss über die Systemleistung.

20:80-Bereitstellungen andererseits führen neue, benutzerdefinierte Dienste ein, die nicht dieselbe Interoperabilität aufweisen, wie 80:20-Bereitstellungen. Aus diesem Grund erstellen Sie einen Prototypen, eine Proof-of-Concept-Bereitstellung, die normalerweise einen härteren Entwicklungs-, Test- und Änderungszyklus durchläuft, bevor die Produktion eingeleitet wird. Mithilfe eines Prototyps können Sie feststellen, wie gut eine geplante Lösung das Problem in einer Testumgebung löst. Wenn der Prototyp zeigt, dass die Funktionalität ausreichend ist, können Sie härtere Tests durchführen und zu einer Pilotbereitstellung übergehen.

---

**Hinweis** – Tatsächlich umgesetzte Unternehmensbereitstellungen können hinsichtlich der Anzahl der erforderlichen benutzerdefinierten Dienste erheblich variieren. In welcher Weise Sie Pilotbereitstellungen und Prototypen für Bereitstellungen zu Testzwecken verwenden, hängt von der Komplexität und der Art Ihrer Bereitstellung ab.

---

---

# Testen von Pilot- und Prototypbereitstellungen

Pilotbereitstellungen und Prototypen für Bereitstellungen werden getestet, um unter Testbedingungen so gut wie möglich zu ermitteln, ob die Bereitstellung die Systemanforderungen und die Geschäftsziele erfüllt.

Im Idealfall sollten Funktionstests beispielhaft für sämtliche gefundenen Verwendungsfälle durchgeführt werden und Messwerte entwickelt werden, anhand derer geprüft werden kann, ob die Bereitstellungen den Anforderungen entsprechen. Funktionstests können auch eine begrenzte Bereitstellung für eine ausgewählte Gruppe von Betabenzern beinhalten, durch die ermittelt wird, ob die Geschäftsanforderungen erfüllt werden.

Mit Belastungstests wird die Leistung bei Spitzenauslastungen getestet. Für derartige Tests wird normalerweise eine Reihe simulierter Umgebungen und Auslastungsgeneratoren verwendet, um den Datendurchsatz und die Leistung zu messen. Die Systemanforderungen für die Bereitstellung sind normalerweise die Grundlage für den Entwurf und die Durchführung von Belastungstests.

---

**Hinweis** – Funktionstests und Stresstests sind besonders wichtig für umfangreiche Bereitstellungen, in denen die Systemanforderungen möglicherweise nicht sorgfältig definiert sind, es keine vorherige Implementierung gibt, die Schätzungen zulässt und die Bereitstellung einen erheblichen Anteil an neuen Entwicklungen erfordert.

---

Tests weisen möglicherweise auf Probleme mit der Bereitstellungskonzeptspezifikation hin und ziehen möglicherweise mehrere Konzept-, Erstellungs- und Testschritte nach sich, bevor die Bereitstellung in einer Produktionsumgebung eingesetzt werden kann. Beim Testen von Prototypen für Bereitstellungen stoßen Sie möglicherweise auf Probleme mit dem Bereitstellungskonzept. In diesem Fall können Sie zu einer früheren Phase im Lebenszyklus der Lösung zurückkehren, um diese Probleme zu beheben.

Stellen Sie sicher, dass Sie Ihr Bereitstellungskonzept gründlich getestet haben, bevor Sie zu einer Pilotbereitstellung übergehen. Eine Pilotbereitstellung setzt voraus, dass Sie das Bereitstellungskonzept bereits durch frühere Tests überprüft haben. Probleme, auf die Sie während des Testens einer Pilotbereitstellung stoßen, müssen im Allgemeinen im Rahmen des Bereitstellungskonzepts behoben werden.

Da eine Produktionsumgebung nie vollständig bei Tests simuliert werden kann und da die Art einer bereitgestellten Lösung sich entwickeln und ändern kann, sollten Sie bereitgestellte Systeme weiterhin überwachen, um sämtliche Bereiche zu identifizieren, für die Optimierung oder Wartung und Service erforderlich sind.

---

## Einsetzen einer Produktionsbereitstellung

Sobald die Pilot- bzw. die Proof-of-Concept-Bereitstellung die Testkriterien erfüllen, können Sie die Bereitstellung in einer Produktionsumgebung einsetzen. Normalerweise gehen Sie hierfür in Etappen vor. Ein etappenweiser Einsatz ist besonders im Fall von umfangreichen Bereitstellungen erforderlich, die viele Benutzer betreffen.

Die etappenweise Bereitstellung kann mit wenigen Benutzern beginnen und Schritt für Schritt auf alle Benutzer ausgeweitet werden. Eine etappenweise Bereitstellung kann auch mit einer begrenzten Anzahl an Diensten begonnen werden und schließlich auf die übrigen Dienste ausgeweitet werden. Wenn Dienste etappenweise in Phasen einbezogen werden, können Probleme isoliert, identifiziert und gelöst werden, auf die ein Dienst in einer Produktionsumgebung stößt.



# Index

---

## Zahlen und Symbole

- 20:80
  - Bereitstellungen, 19
  - Implementierungsphase, 105
- 3-dimensionale Architektur, 50
- 80:20
  - Bereitstellungen, 19, 105

## A

- Access Manager, 53, 78
- Anforderungen an die Dienstebene, 48
  - Anforderungen, 48
- Anwendungsanalyse, 36-38
  - Aswirkungen auf das
    - Bereitstellungskonzept, 72
- Anwendungsfälle, 38-39
  - Auswirkungen auf das
    - Bereitstellungskonzept, 72
  - Einschätzen von
    - Prozessoranforderungen, 78-79
  - identitätsbasiertes
    - Kommunikationsbeispiel, 64-67
    - Messaging Server-Beispiel, 60-63
- Anwendungsmuster, 30
- Application Server, 53

## B

- Beispiele
  - Bereitstellungsarchitektur, 100-101

## Beispiele (Fortsetzung)

- Directory Server, 91
  - Einschätzen von
    - Prozessoranforderungen, 75-80
  - Einschätzen von Prozessoranforderungen für sichere Transaktionen, 82-83
  - Einzelmaster-Replikation, 91-92
  - Failover, 90-91
  - Identitätsbasierte Kommunikation, 63-67
  - Lastenausgleich, 86, 88-90
  - logische Architektur, 58-67
  - Messaging Server, logische
    - Architektur, 58-63
  - Multimaster-Replikation, 92-93
  - Replikation von Diensten, 91
  - Skalierbarkeit, 94-95
  - Verfügbarkeitskonzept, 88-93
  - Zugriffszonen, 67
- Belastungstests, 106
  - Benutzerverwaltungsplan, 71
  - Bereitstellungsarchitektur, 70-71
    - Beispiel, 100-101
  - Bereitstellungskonzept
    - Ausgaben, 70-71
    - Faktoren, 72-73
    - Informationen, 69-73
    - Methodik, 73-74
    - Projektgenehmigung, 70
  - Bereitstellungsplanung
    - Informationen, 21-26
    - Lösungslebenszyklus, 21-23
    - schrittweise Annäherung, 32
  - Bereitstellungsszenario, 49, 68, 69

Betriebsanforderungen, 30  
Betriebsphase, 26  
Betriebsplan (Run Book), 71  
Budgeteinschränkungen, 34

## C

Calendar Server, 53, 78  
Client-Schicht, mehrschichtige Architektur, 57  
Communications Express, 53

## D

Datenschicht, mehrschichtiges  
Architekturmodell, 57  
Dienstqualitätsanforderungen, 39-48  
Auswirkungen auf das  
Bereitstellungskonzept, 72  
Funktion im Bereitstellungskonzept, 69  
Directory Proxy Server, 53, 56  
Directory Server, 53, 59, 78  
Einzelmaster-Replikation, 91  
Multimaster-Replikation, 91, 92-93  
DMZ, externe Zugriffszone, 68  
Dokumentation  
Installation Guide, 53  
Installationshandbuch, 104  
Technischer Überblick, 19, 50, 51  
Dreidimensionale Architektur, 50

## E

Einschätzen von Prozessoranforderungen, 73,  
74-80  
Anwendungsfälle, 78-79  
Beispiel, 75-80  
sichere Transaktionen, 81-84  
Einzelmaster-Replikation, 91  
Beispiel, 91-92  
Externe Zugriffszone (DMZ), 68

## F

Failover, 85

Failover (Fortsetzung)  
Beispiel, 90-91  
Sun Cluster-Software, 88  
Fehlertolerante Systeme, 42  
Funktionstests, 106

## G

Gesamtkosten, 34  
Auswirkungen auf das  
Bereitstellungskonzept, 72  
Geschäftliche Einschränkungen, 33-34  
Budgeteinschränkungen, 34  
Gesamtkosten, 34  
Migrationsprobleme, 33  
Planungsmandate, 33  
Geschäftsanforderungen  
Anwendungsmuster, 30  
Betriebsanforderungen, 30  
definieren, 28-33  
Geschäftsziele, 28-29  
Kenntnisse über Benutzer, 29-30  
Regulierungsanforderungen, 31  
Sicherheitsziele, 31  
Unternehmenskultur, 30-32  
Vereinbarungen auf Dienstebene, 32-33  
Geschäftsdienstschicht, mehrschichtiges  
Architekturmodell, 57  
Geschäftsziele  
Auswirkungen auf das  
Bereitstellungskonzept, 73  
festlegen, 28-29  
Glossar, Verweis auf, 13

## H

Horizontal redundante Systeme, 86-88

## I

Identifizieren von Engpässen.,  
Bereitstellungskonzept, 74  
Identitätsbasierte Kommunikationslösung,  
Beispiel, Einschätzen von  
Prozessoranforderungen, 75

- Identitätsbasiertes
  - Kommunikationsbeispiel, 63-67
  - Anwendungsfälle, 64-67
- Implementierungsphase, 25-26, 105
  - Entwickeln von Pilotbereitstellungen und Prototypen, 105
  - Informationen, 103-104
- Implementierungspläne, 70-71
- Implementierungsspezifikationen, 70-71
- Installation von Java Enterprise System, 104
- Installationsplan, 71
- Instant Messaging, 53
- Interne Zugriffszone (Intranet), 68

## J

- Java Enterprise System
  - 20:80
    - Bereitstellungen, 19
  - 80:20
    - Bereitstellungen, 19
  - benutzerdefinierte Dienste, 19-20
  - Dienste, 19-20
  - dreidimensionale Architektur, 50
  - Einsetzen einer
    - Produktionsbereitstellung, 107
  - Informationen, 17
  - installieren, 104
  - Komponenten, 51-58
  - Komponentenabhängigkeiten, 52-55
  - Migrationsprobleme, 20-21
  - Systemdienste, 17-19
  - Zugriffskomponenten, 56

## K

- Komponentenabhängigkeiten, 52-55
  - Webcontainer-Unterstützung, 55

## L

- Lastenausgleich, 85
  - Beispiel, 86
- Latente Kapazität, 46

- Latente Kapazität (Fortsetzung)
  - Überlegungen hinsichtlich Skalierbarkeit, 93-94
- Leistung
  - Dienstqualitätsanforderung, 40-41
  - Identifizieren von Engpässen, 95-98
  - Optimieren von Ressourcen, 98
- Logische Architektur
  - Auswirkungen auf das
    - Bereitstellungskonzept, 72-73
  - Beispiele, 58-67
  - Entwurf, 51
  - identitätsbasiertes
    - Kommunikationsbeispiel, 63
- Logische Architekturen, 49-51
- Logische Schichten, mehrschichtiges Architekturmodell, 57
- Logisches Konzept, Informationen, 49-51
- Lösungslebenszyklus, 21-23
  - Bereitstellungskonzeptphase, 69-73
  - Betriebsphase, 26
  - Implementierungsphase, 25-26, 103-104
  - Phase der Geschäftsanalyse, 23, 27
  - Phase der technischen
    - Anforderungen, 23-24, 35-36
  - Phase des Bereitstellungskonzepts, 24-25
  - Phase des logischen Konzepts, 24, 49-51

## M

- Mehrschichtiges Architekturmodell, 56-58
- Message Queue, 53
- Messaging Server, 53
  - Anwendungsfälle, 60-63
  - Beispiel für Lastenausgleich, 88-90
  - logische Architektur, Beispiel, 58-63
  - logische eindeutige Dienste, 55-56
  - Message Multiplexor (MMP), 56, 59, 78
  - Message Store (Nachrichtenspeicher), 78
  - Message Store (STR), 55
  - Message Transfer Agent (MTA), 55, 59
  - Messenger Express Multiplexor (MEM), 56
  - Nachrichtenspeicher (STR), 59
- Migrationsplan, 71
- Migrationsprobleme, 20-21
  - als geschäftliche Einschränkung, 33
- Multimaster-Replikation, 91

Multimaster-Replikation (Fortsetzung)  
Beispiel, 92-93

## **N**

N+1-Failover-System, 86-88

## **O**

Optimieren  
Festplattenzugriff, 97-98  
Ressourcennutzung, 98-99  
Optimieren von Ressourcen,  
Bereitstellungskonzept, 74

## **P**

Phase der Geschäftsanalyse, 23  
Informationen, 27  
Phase der technischen Anforderungen, 23-24  
Anwendungsanalyse, 36-38  
Anwendungsfälle, 38-39  
Dienstqualitätsanforderungen, 39-48  
Informationen, 35-36  
Phase des Bereitstellungskonzepts, 24-25  
Phase des logischen Konzepts, 24  
Pilotbereitstellungen, 105  
testen, 106  
Planungsmandate, 33  
Portal Server, 53, 56  
Mobile Access, 56  
Secure Remote Access, 53, 56  
Präsentationsschicht, mehrschichtiges  
Architekturmodell, 57  
Projektgenehmigung, 70  
Prototypen, 105  
testen, 106  
Prozessoranforderungen, einschätzen, 74-80

## **Q**

QoS (Qualität of Service Requirements), 39-48

## **R**

Regulierungsanforderungen, 31  
Replikation von Diensten  
Directory Server-Beispiel, 91  
Verfügbarkeitsstrategie, 85  
Replizieren von Diensten, 73  
Risikoverwaltung, 99-100  
Roll-out-Plan, 71  
Run Book, 71

## **S**

Schulungsbuch, 71  
Sichere Zugriffszone, 68  
Sicherheit  
Dienstqualitätsanforderungen, 45-46  
Einschätzen von  
Prozessoranforderungen, 73  
Optimieren von Ressourcen, 99  
Skalierbarkeit  
Beispiel, 94-95  
Dienstqualitätsanforderung, 43-45  
Optimieren von Ressourcen, 99  
Strategien, 93-95  
Wachstumseinschätzung, 44-45  
SLA, 32-33  
Sun Cluster-Software, 88  
Failover-Beispiel, 90-91

## **T**

Technische Anforderungen  
Dienstqualitätsanforderungen, 48  
Latente Kapazität, 46  
Leistung, 40-41  
Sicherheit, 45-46  
Skalierbarkeit, 43-45  
Verfügbarkeit, 41-43  
Wartungseignung, 47-48  
Testplan, 71  
Tests  
Belastungstests, 106  
Funktionstests, 106  
Pilotbereitstellungen und Prototypen, 106



## **U**

Unternehmenskultur, 30-32

## **V**

Vereinbarungen auf Dienstebene, 32-33

    Auswirkungen auf das  
    Bereitstellungskonzept, 72

Verfügbarkeit

    Beispiele, 88-93  
    Dienstqualitätsanforderungen, 41-43  
    Failover, 85  
    horizontal redundante Systeme, 86-88  
    Lastenausgleich, 85  
    N+1-Failover-System, 86-88  
    Optimieren von Ressourcen, 99  
    Priorisieren, 42-43  
    Replikation von Diensten, 85

Verfügbarkeitsstrategien, ermitteln, 84-93

Verwalten von Risiken, 99-100  
    Bereitstellungskonzept, 74

## **W**

Wartungseignung

    Dienstqualitätsanforderung, 47-48  
    Optimieren von Ressourcen, 99

Web Server, 53, 78

Wiederherstellungsplan, 71

## **Z**

Zugriffszonen, 67-68

