



Sun Java™ System

Sun Java Enterprise System 2005Q1 Handbuch zur Bereitstellungsplanung

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
USA

Teilenr.: 819-1916

Copyright © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, USA. Alle Rechte vorbehalten.

Sun Microsystems, Inc. hat Rechte in Bezug auf geistiges Eigentum an der Technologie, die in dem in diesem Dokument beschriebenen Produkt enthalten ist. Im Besonderen und ohne Einschränkung umfassen diese Ansprüche in Bezug auf geistiges Eigentum eines oder mehrere der unter <http://www.sun.com/patents> aufgelisteten Patente und eines oder mehrere Patente oder Anwendungen mit laufendem Patent in den USA und in anderen Ländern.

DIESES PRODUKT ENTHÄLT VERTRAULICHE INFORMATIONEN UND GESCHÄFTSGEHEIMNISSE VON SUN MICROSYSTEMS, INC. VERWENDUNG, OFFENLEGUNG ODER REPRODUKTION IST OHNE VORHERIGE SCHRIFTLICHE GENEHMIGUNG DURCH SUN MICROSYSTEMS, INC., NICHT GESTATTET.

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

Diese Ausgabe kann von Drittanbietern entwickelte Bestandteile enthalten.

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, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, das Duke-Logo, das Java-Kaffeetassenlogo, das Solaris-Logo, das SunTone Certified-Logo und das Sun ONE-Logo 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 SPARC-Warenzeichen aufweisen, basieren auf der von Sun Microsystems, Inc., entwickelten Architektur.

Legato und das Legato-Logo sind eingetragene Warenzeichen und Legato NetWorker ist ein Warenzeichen bzw. ein eingetragenes Warenzeichen von Legato Systems, Inc. Das Netscape Communications Corp-Logo ist ein Warenzeichen bzw. ein eingetragenes Warenzeichen der Netscape Communications Corporation.

Die grafische Benutzeroberfläche von OPEN LOOK und Sun™ wurden von Sun Microsystems, Inc., entwickelt, für die entsprechenden Benutzer und Lizenznehmer. 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-Benutzeroberflächen implementieren und sich ansonsten an die schriftlichen Sun-Lizenzvereinbarungen halten.

Produkte, die in diesem Diensthandbuch beschrieben sind, und die in diesem Handbuch enthaltenen Informationen unterliegen den Gesetzen der US-Exportkontrolle und können den Export- oder Importgesetzen anderer Länder unterliegen. Die Verwendung im Zusammenhang mit Nuklearwaffen, Raketenwaffen, chemischen und biologischen Waffen, im nuklear-maritimen Bereich oder durch in diesem Bereich tätige Endbenutzer, direkt oder indirekt, ist strengstens untersagt. Der Export oder Rückexport in Länder, die einem US-Embargo unterliegen, oder an Personen und Körperschaften, die auf der US-Exportausschlussliste stehen, einschließlich (jedoch nicht beschränkt auf) der Liste nicht zulässiger Personen und speziell ausgewiesener Staatsangehöriger, ist strengstens untersagt.

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

Inhalt

Liste der Abbildungen	7
Liste der Tabellen	9
Vorwort	11
An wen richtet sich dieses Handbuch	12
Aufbau dieses Handbuchs	12
Verwandte Dokumentation	12
Handbücher in diesem Dokumentationsatz	12
Online-Zugriff auf Sun-Ressourcen	14
Kontakt zum technischen Support von Sun	15
Verwandte Website-Referenzen von Drittanbietern	15
Kommentare sind willkommen	15
Kapitel 1 Einführung in die Bereitstellungsplanung	17
Informationen zu Java Enterprise System	17
Systemdienste	18
Integrierte Dienste und benutzerdefinierte Dienste	19
Migration auf Java Enterprise System	21
Informationen zur Bereitstellungsplanung	21
Lebenszyklus einer Lösung	22
Phase der Geschäftsanalyse	23
Phase der technischen Anforderungen	23
Phase des logischen Konzepts	24
Phase des Bereitstellungskonzepts	25
Implementierungsphase	26
Betriebsphase	27

Kapitel 2 Geschäftsanalyse	29
Informationen zur Geschäftsanalyse	29
Definieren der Geschäftsanforderungen	30
Festlegen der Geschäftsziele	30
Bereich	30
Prioritäten	30
Entscheidende Qualitäten	31
Wachstumsfaktoren	31
Sicherheitsspielraum	31
Kenntnis der Benutzerbedürfnisse	31
Erarbeiten von Betriebsanforderungen	32
Unterstützung vorhandener Anwendungsmuster	33
Kenntnis der Unternehmenskultur	33
Interessengruppen	33
Standards und Richtlinien	34
Regulierungsanforderungen	34
Sicherheit	34
Standortverteilung	35
Schrittweise Annäherung	35
Kenntnis der Vereinbarungen auf Dienstebene	36
Definieren der geschäftlichen Einschränkungen	36
Migrationsprobleme	37
Planungsmandate	37
Budgeteinschränkungen	37
Gesamtkosten	38
Kapitel 3 Technische Anforderungen	39
Informationen über technische Anforderungen	39
Anwendungsanalyse	40
Anwendungsfälle	42
Dienstqualitätsanforderungen	43
Leistung	45
Verfügbarkeit	46
Fehlertolerante Systeme	46
Priorisieren der Dienstverfügbarkeit	47
Dienstausfall	48
Skalierbarkeit	48
Wachstumseinschätzung	49
Sicherheitsanforderungen	50
Elemente eines Sicherheitsplans	51
Latente Kapazität	52
Wartungseignungsanforderungen	52
Anforderungen an die Dienstebene	53

Kapitel 4 Logisches Konzept	55
Informationen zu logischen Architekturen	55
Entwurf einer logischen Architektur	57
Java Enterprise System-Komponenten	58
Komponentenabhängigkeiten	59
Webcontainer-Unterstützung	62
Von Messaging Server bereitgestellte logisch eindeutige Dienste	62
Zugriffskomponenten	63
Entwurf einer mehrschichtigen Architektur	64
Beispiele für logische Architekturen	65
Messaging Server-Beispiel	66
Messaging Server-Anwendungsfälle	67
Identitätsbasiertes Kommunikationsbeispiel	71
Anwendungsfälle für identitätsbasiertes Kommunikationsbeispiel	72
Zugriffszonen	75
Bereitstellungsszenario	76
Kapitel 5 Bereitstellungskonzept	77
Bereitstellungskonzept	78
Projektgenehmigung	78
Bereitstellungskonzeptausgaben	79
Faktoren mit Auswirkung auf das Bereitstellungskonzept	81
Bereitstellungskonzept Methodik	82
Einschätzen von Prozessoranforderungen	83
Beispiel für das Einschätzen von Prozessoranforderungen	85
Ermitteln der CPU-Basiseinschätzung für Benutzereinstiegspunkte	86
CPU-Einschätzungen für Dienstabhängigkeiten aufnehmen	87
Anwendungsfälle auf Spitzenauslastung hin prüfen	88
Einschätzungen für andere Ladebedingungen ändern	89
CPU-Einschätzungen aktualisieren	89
Einschätzen von Prozessoranforderungen für sichere Transaktionen	90
CPU-Einschätzungen für sichere Transaktionen	91
Spezielle Hardware zur Handhabung von SSL-Transaktionen	93
Ermitteln von Verfügbarkeitsstrategien	94
Verfügbarkeitsstrategien	94
System mit nur einem Server	95
Horizontal redundante Systeme	96
Sun Cluster-Software	98
Beispiele für Verfügbarkeitskonzepte	98
Beispiel des Lastenausgleichs für Messaging Server	98
Failover-Beispiel mit Sun Cluster-Software	100
Beispiel für Replikation von Verzeichnisdiensten	101
Einzelmaster-Replikation	101
Multimaster-Replikation	102

Ausarbeiten von Strategien hinsichtlich der Skalierbarkeit	103
Latente Kapazität	104
Skalierbarkeitsbeispiel	105
Identifizieren von Leistungsgaps	106
Optimieren des Festplattenzugriffs	108
Konzeption für optimale Ressourcennutzung	109
Verwalten von Risiken	111
Beispiel für eine Bereitstellungsarchitektur	112
Kapitel 6 Implementierung eines Bereitstellungskonzepts	115
Informationen zur Implementierung des Bereitstellungskonzepts	116
Installation und Konfiguration der Software	117
Entwickeln von Pilotbereitstellungen und Prototypen	117
Testen von Pilot- und Prototypbereitstellungen	118
Einsetzen einer Produktionsbereitstellung	120
Index	121

Liste der Abbildungen

Abbildung 1-1 Lebenszyklus einer Lösung	22
Abbildung 4-1 Die drei Pfeiler der Java Enterprise System-Lösungsarchitektur	56
Abbildung 4-2 Java Enterprise System-Komponenten	58
Abbildung 4-3 Java Enterprise System-Komponentenabhängigkeiten	61
Abbildung 4-4 Mehrschichtiges Architekturmodell	64
Abbildung 4-5 Logische Architektur für die Messaging Server-Bereitstellung	66
Abbildung 4-6 Logische Architektur von Messaging Server, in der Anwendungsfall 1 dargestellt wird	68
Abbildung 4-7 Logische Architektur von Messaging Server, in der Anwendungsfall 2 dargestellt wird	69
Abbildung 4-8 Logische Architektur von Messaging Server, in der Anwendungsfall 3 dargestellt wird	70
Abbildung 4-9 Logische Architektur für identitätsbasiertes Kommunikationsszenario	72
Abbildung 4-10 Logische Architektur des Kommunikationsszenarios, in der Anwendungsfall 1 dargestellt wird	73
Abbildung 4-11 Logische Architektur des Kommunikationsszenarios, in der Anwendungsfall 2 dargestellt wird	74
Abbildung 4-12 In Zugriffszonen platzierte logische Komponenten	75
Abbildung 5-1 Logische Architektur für ein identitätsbasiertes Kommunikationsszenario	86
Abbildung 5-2 System mit nur einem Server	95
Abbildung 5-3 n+1-Failover-System mit zwei Servern	96
Abbildung 5-4 Lastenausgleich plus Failover zwischen zwei Servern	96
Abbildung 5-5 Verteilung der Last zwischen n Servern	97
Abbildung 5-6 Verfügbarkeitsstrategie für Messaging Server-Beispiel	99
Abbildung 5-7 Failover-Konzept mit Sun Cluster-Software	100
Abbildung 5-8 Beispiel einer Einzelmaster-Replikation	101
Abbildung 5-9 Beispiel einer Multimaster-Replikation	103
Abbildung 5-10 Beispiele für horizontale und vertikale Skalierung	106
Abbildung 5-11 Beispiel für eine Bereitstellungsarchitektur	113

Liste der Tabellen

Tabelle 1	Java Enterprise System Dokumentation	13
Tabelle 1-1	Java Enterprise System Dienstkategorien	19
Tabelle 3-1	Faktoren für die Anwendungsanalyse	41
Tabelle 3-2	Systemqualitäten mit Auswirkung auf dieQoS-Anforderungen	44
Tabelle 3-3	Unvorhergesehene Ausfallzeiten für ein System, das das ganze Jahr (8760 Stunden) läuft	46
Tabelle 3-4	Verfügbarkeit von Diensten nach Priorität	47
Tabelle 3-5	Skalierbarkeitsfaktoren	49
Tabelle 3-6	Aspekte für die Wartungseignungsanforderungen	52
Tabelle 4-1	Java Enterprise System Komponentenabhängigkeiten	59
Tabelle 4-2	Messaging Server-Konfigurationen	62
Tabelle 4-3	Java Enterprise System Komponenten, die Remote-Zugriff bieten	63
Tabelle 4-4	Logische Schichten in einer mehrschichtigen Architektur	64
Tabelle 4-5	Komponenten in der logischen Architektur von Messaging Server	67
Tabelle 4-6	Sichere Zugriffszonen und darin platzierte Komponenten	76
Tabelle 5-1	CPU-Einschätzungen für Komponenten mit zugriffsbezogenen Benutzereinstiegspunkten	87
Tabelle 5-2	CPU-Einschätzungen für unterstützende Komponenten	87
Tabelle 5-3	Anpassungen der CPU-Einschätzung hinsichtlich der Spitzenauslastung	88
Tabelle 5-4	Anpassungen von CPU-Einschätzungen für unterstützende Komponenten	90
Tabelle 5-5	Ändern von CPU-Einschätzungen für sichere Transaktionen	92
Tabelle 5-6	Anpassungen der CPU-Einschätzung für sichere Portal Server-Transaktionen	93
Tabelle 5-7	Anpassungen von CPU-Einschätzungen für unterstützende Komponenten	98
Tabelle 5-8	Datenzugriffspunkte	107
Tabelle 5-9	Überlegungen in Bezug auf die Ressourcenverwaltung	110

Vorwort

Das Handbuch *Java Enterprise System – Handbuch zur Bereitstellungsplanung* bietet eine Einführung in die Planung und 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 hochklassige Beispiele und Strategien angegeben, die Sie bei der Planung der unternehmensweiten Bereitstellung von Lösungen mit Java Enterprise System anwenden können.

Dieses Vorwort enthält die folgenden Abschnitte:

- „An wen richtet sich dieses Handbuch“ auf Seite 12
- „Verwandte Dokumentation“ auf Seite 12
- „Online-Zugriff auf Sun-Ressourcen“ auf Seite 14
- „Kontakt zum technischen Support von Sun“ auf Seite 15
- „Verwandte Website-Referenzen von Drittanbietern“ auf Seite 15
- „Kommentare sind willkommen“ auf Seite 15

Lesen Sie, bevor Sie irgendwelche in diesem Handbuch beschriebenen Aufgaben durchführen die *Java Enterprise System Release Notes* (<http://docs.sun.com/doc/819-0812>) aufmerksam durch.

An wen richtet sich dieses Handbuch

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

In dem vorliegenden Leitfaden wird angenommen, dass Sie mit dem Konzept und der Installation von Unternehmensanwendung und dem Inhalt des Handbuchs *Java Enterprise System Technical Overview* vertraut sind.

Aufbau dieses Handbuchs

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

Verwandte Dokumentation

Auf der Website <http://docs.sun.com>SM können Sie online auf die technische Dokumentation von Sun zugreifen. Sie können das Archiv durchsuchen oder nach einem bestimmten Handbuchtitel oder Thema suchen.

Handbücher in diesem Dokumentationsatz

Die Java ES-Handbücher sind als Online-Dateien im PDF-Format (Portable Document Format) und HTML-Format (Hypertext Markup Language) verfügbar. Beide Formate können von unterstützenden Technologien für Benutzer mit Behinderungen gelesen werden. Unter folgender Adresse können Sie auf die SunTM-Dokumentations-Website zugreifen:

<http://docs.sun.com>

Die Java ES-Dokumentation enthält Informationen über das System in seiner Gesamtheit sowie über seine Komponenten. Zugriff auf diese Dokumentation:

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

In der nachfolgenden Tabelle sind die Handbücher auf Systemebene aufgeführt, die Bestandteil des Java ES-Dokumentationssatzes sind. In der linken Spalte werden der Name sowie der Speicherort der Teilenummer der einzelnen Dokumente angegeben. In der rechten Spalte wird der allgemeine Inhalt des Dokuments beschrieben.

Tabelle 1 Java Enterprise System Dokumentation

Dokument	Inhalt
<i>Java Enterprise System Release Notes</i> http://docs.sun.com/doc/819-0812	Enthält die neuesten Informationen über das Java Enterprise System, einschließlich bekannter Probleme. Zudem verfügen die Komponenten über ihre eigenen Versionshinweise.
<i>Java Enterprise System Documentation Roadmap</i> http://docs.sun.com/doc/819-1909	Enthält Beschreibungen der Dokumentation zu Java Enterprise System. Es sind Verknüpfungen zur Dokumentation für die Komponenten enthalten.
<i>Java Enterprise System Technical Overview</i> http://docs.sun.com/doc/819-1923	Bietet eine Einführung in die technischen und konzeptionellen Grundlagen von Java Enterprise System. Beschreibt die Komponenten, die Architektur sowie Prozesse und Funktionen.
<i>Java Enterprise System – Handbuch zur Bereitstellungsplanung</i> http://docs.sun.com/doc/819-1916	Bietet eine Einführung in die Planung und die Konzeptentwicklung von Bereitstellungslösungen für Unternehmen, die auf Java Enterprise System basieren. Beschreibt die Grundlagen und Prinzipien der Bereitstellungsplanung und des Bereitstellungskonzepts sowie den Lebenszyklus einer Lösung und bietet hochklassige Beispiele und Strategien, die Sie bei der Planung der unternehmensweiten Bereitstellung von Lösungen mit Java Enterprise System anwenden können.
<i>Sun Java Enterprise System User Management Guide</i> http://docs.sun.com/doc/817-5761	Unterstützt Sie bei der Planung, Bereitstellung und Verwaltung von Informationen über die Benutzer Ihrer Java Enterprise System-Lösung. Stellt durch die Beschreibung von Benutzerverwaltungsaufgaben in den einzelnen Phasen des Lebenszyklus einer Lösung eine Ergänzung zum Handbuch <i>Java Enterprise System – Handbuch zur Bereitstellungsplanung</i> dar.
<i>Java Enterprise System Bereitstellungsbeispiele: Bewertungsszenario</i> http://docs.sun.com/doc/819-0059	Beschreibt die Installation von Java Enterprise System auf einem System, das Herstellen einer Reihe wichtiger freigegebener Dienste und Netzwerkdienste und die Einrichtung der Benutzerkonten, mit denen auf die von Ihnen eingerichteten Dienste zugegriffen werden kann.

Tabelle 1 Java Enterprise System Dokumentation (*Fortsetzung*)

Dokument	Inhalt
<i>Java Enterprise System-Installationshandbuch</i> http://docs.sun.com/doc/819-0805	Führt Sie durch die Installation von Java Enterprise System für das Solaris™- oder das Linux-Betriebssystem. Zeigt auf, wie Sie die zu installierenden Komponenten auswählen, wie Sie diese Komponenten nach der Installation konfigurieren und wie Sie überprüfen, dass die konfigurierten Komponenten ordnungsgemäß funktionieren.
<i>Java Enterprise System Upgrade and Migration Guide</i> http://docs.sun.com/doc/819-0062	Bietet Informationen und Anweisungen für die Aufrüstung von Java Enterprise System für die Solaris™- oder die Linux-Betriebsumgebung.
<i>Java Enterprise System Glossary</i> http://docs.sun.com/doc/816-1930	Definiert die Begriffe, die in der Java Enterprise System-Dokumentation verwendet werden.

Online-Zugriff auf Sun-Ressourcen

Produktdownloads, professionelle Dienste, Patches und Support sowie zusätzliche Informationen für Entwickler erhalten Sie hier:

- Download Center
<http://www.sun.com/software/download/>
- Professionelle Dienste
<http://www.sun.com/service/sunps/sunone/index.html>
- Sun Enterprise Services, Solaris-Patches und Support
<http://sunsolve.sun.com/>
- Informationen für Entwickler
<http://developers.sun.com/prodtech/index.html>

Unter folgender Adresse erhalten Sie Informationen über Java ES und seine Komponenten:

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

Kontakt zum technischen Support von Sun

Wenn Sie technische Fragen zu diesem Produkt haben, die nicht in der Produktdokumentation beantwortet werden, besuchen Sie die Website <http://www.sun.com/service/contacting>.

Verwandte Website-Referenzen von Drittanbietern

Sun ist nicht haftbar für die Verfügbarkeit der Websites Dritter, die in diesem Dokument erwähnt werden. 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 „Send Comments“. 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 am Anfang des Dokuments finden.

Kommentare sind willkommen

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 Lösungszyklus, 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, mit einer allgemeinen Gruppe freigegebener Bibliotheken synchronisiert und verwenden alle ein integriertes System für die Benutzeridentitäts- und Sicherheitsverwaltung.

Systemdienste

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

- **Zugangsdienste:** 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 freigegebenen Komponenten die Kommunikation und unterstützen die Entwicklung, Bereitstellung und Verwaltung von Anwendungen für unterschiedliche 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 Portal Server Secure Remote Access Access Manager Directory Server Application Server oder Web Server
Kommunikations- und 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 architektonischen Konzepten von Java Enterprise Systemerhalten Sie im *Java Enterprise System Technical Overview*, <http://docs.sun.com/doc/819-0061>.

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 zur Verfügung.
- **20:80-Bereitstellungen:** Diese Lösungen bestehen aus zahlreichen benutzerdefinierten Diensten und Drittanbieteranwendungen.

Bei den Kategorien 80:20 und 20:80 handelt es sich um grobe Verallgemeinerungen. 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 [Tabelle 1-1 auf Seite 19](#) aufgelisteten Dienstkategorien können für die Verteilung von 80:20-Bereitstellungen verwendet werden. Kommunikations- und Zusammanarbeitdienste stellen den Endbenutzern beispielsweise E-Mail-, Kalender- und Instant Messaging-Funktionen zur Verfügung, mit denen Sie 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 die Planung, der Entwurf und die Implementierung hauptsächlich von den speziellen Anforderungen Ihres Unternehmens bestimmt. Lösungen für die erstmalige Bereitstellung sind jedoch nicht die Regel. Wahrscheinlicher ist es, dass Java Enterprise System verwendet wird, 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 Upgrade and Migration Guide*.

Informationen zur Bereitstellungsplanung

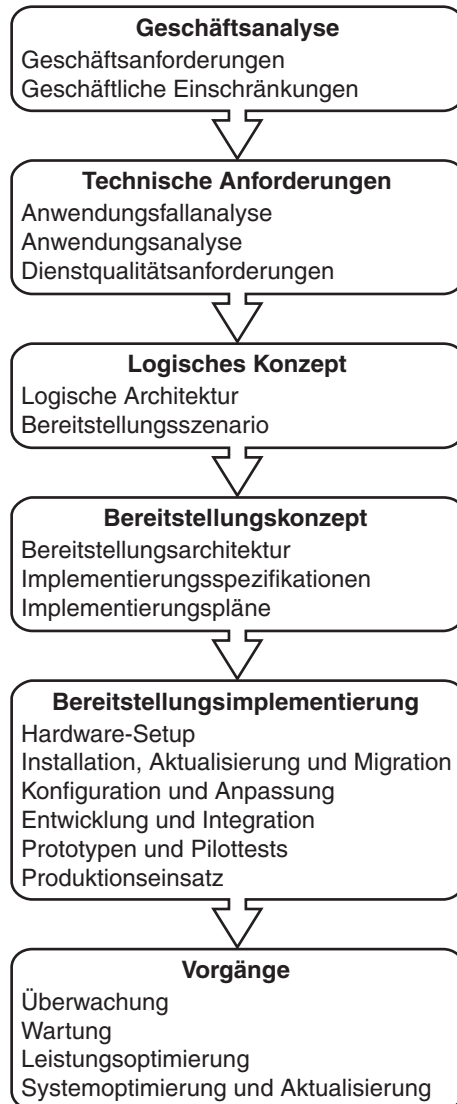
Die Bereitstellungsplanung ist ein entscheidender Schritt bei der erfolgreichen Implementierung einer Java Enterprise System-Lösung. Jedes Unternehmen verfügt über eigene Ziele, Anforderungen und Prioritäten, die es zu berücksichtigen gilt. Die erfolgreiche Planung beginnt mit der Analyse der Ziele eines 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 aufeinanderfolgenden Phasen. Jede Phase umfasst entsprechende Aufgaben, deren Ergebnisse jeweils als Ausgangspunkt für die darauffolgenden Phasen dienen. Die Aufgaben innerhalb der einzelnen Phasen sind von Wiederholungen geprägt und erfordern sorgfältige Analyse- und Entwurfsmaßnahmen, bevor entsprechende Ergebnisse für die jeweilige Phase generiert werden können. Auch in den frühen Phasen können Wiederholungen auftreten. Während der Phase des Bereitstellungs-konzepts 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](#), „Geschäftsanalyse“ auf Seite 29.

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 (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 Lösungszyklus. 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 nachfolgende Erstellung der Vereinbarungen auf Dienstebene (Service Level Agreement, SLA) bilden, festlegen. In einer Vereinbarung auf Dienstebene (Service Level Agreement, SLA) werden die Bedingungen angegeben, unter denen die Kundenunterstützung zur Verfügung gestellt werden muss, damit das System ordnungsgemäß gewartet werden kann. Sie sind im Allgemeinen als Bestandteil der Projektgenehmigung in der Phase des Bereitstellungskonzepts aufgeführt.

Weitere Informationen zu den technischen Anforderungen erhalten Sie in [Kapitel 3, „Technische Anforderungen“](#) auf Seite 39.

Phase des logischen Konzepts

Während der Erstellung des logischen Konzepts, bei der die Anwendungsfälle aus der Phase der technischen Anforderungen als Ausgangspunkt verwendet werden, 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](#), „[Logisches Konzept](#)“ auf Seite 55.

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 der oberen 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 Bereitstellungskonzeptphase 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. Implementierungspläne enthalten einen Migrationsplan, einen Installationsplan, einen Benutzerverwaltungsplan, einen Testplan und sonstige Pläne.

Weitere Informationen zum Bereitstellungs-konzept erhalten Sie in [Kapitel 5](#), „Bereitstellungskonzept“ auf Seite 77.

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 eines Benutzerverwaltungsplans
- Erstellung einer Pilot- und einer Prototypimplementierung 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

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

Weitere Informationen zur Implementierungsphase erhalten Sie in [Kapitel 6](#), „Implementierung eines Bereitstellungs-konzepts“ auf Seite 115.

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 29
- „[Definieren der Geschäftsanforderungen](#)“ auf Seite 30
- „[Definieren der geschäftlichen Einschränkungen](#)“ auf Seite 36

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 sein müssen. Berücksichtigen Sie auch sämtliche unternehmerischen Einschränkungen, die die Erreichung dieser Ziele erschweren/hindern/beeinträchtigen. Am Ende der Analyse der Geschäftsanforderungen und -einschränkungen haben Sie eine Gruppe von Dokumenten mit Geschäftsanforderungen in der Hand.

Diese Reihe an Dokumenten mit 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

Es gibt keine einfache Formel, mit der alle Geschäftsanforderungen bestimmt werden können. Sie bestimmen die Anforderungen basierend auf der Zusammenarbeit mit den Interessengruppen, von denen eine Softwarelösung benötigt wird, Ihrer Kenntnis der Geschäftsdomäne und angewandtem kreativem 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 Bereitstellungs-konzept 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 können Sie Richtlinien für Entscheidungen angeben, die möglicherweise getroffen werden müssen, damit Ihr Bereitstellungs-konzept 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 ändern könnten und wie sie sich über längere Zeiträume entwickeln. 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 Arten von Benutzern 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 aktuelle Mitarbeiter
- Aktuelle 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 für Vorteile aufgeführt, die Benutzern durch eine Lösung entstehen können:

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

Erarbeiten von Betriebsanforderungen

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

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

Drücken Sie die Betriebsanforderungen in messbaren Begriffen aus, die alle Interessengruppen verstehen. Vermeiden Sie nicht eindeutige Formulierungen, wie beispielsweise „adäquate Endbenutzerantwortzeit“. Mögliche Beispiele für die Betriebsanforderungen:

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

Unterstützung vorhandener Anwendungsmuster

Drücken Sie vorhandene Anwendungsmuster als klar messbare Ziele aus. 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 durch?

Beobachten Sie die Benutzer, die auf Ihre Dienste zugreifen. Wann und für wie lange die Benutzer auf die vorhandenen Dienste zugreifen sind Faktoren, die bei der Ermittlung Ihrer Ziele helfen. 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, können die Pläne 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 Fahrplan 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 die Speicherung persönlicher Daten.

Sicherheit

Einige Ziele, die Sie ermitteln, bringen möglicherweise Sicherheitsprobleme mit sich, auf die besonders viel Wert gelegt werden sollte. Heben Sie bestimmte Sicherheitsziele hervor, die für die Lösung besonders wichtig sind. Beispielsweise:

- Autorisierter Zugriff auf vertrauliche Informationen
- Funktionsbasierter/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 negativ 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 schrittweise.

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 zunächst zurückgestellt wird.

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 Bereitstellungsimpementierung nutzen.
- Sie können Anforderungen in Bezug auf die Investitionsausgaben miteinbeziehen.
- 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 vonnöten ist. Eine Vereinbarung auf Dienstebene basiert auf den Geschäftsanforderungen, die bei der Geschäftsanalyse definiert werden. Später, in der auf die technischen Anforderungen bezogenen Phase, 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 in ihr 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 vermutlich Faktoren sind, die sich auf das Erreichen des Geschäftsziels auswirken, 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 auch in Betracht gezogen werden sollten. Als Meilensteine können interne Ereignisse, wie beispielsweise geplante Dienstrollouts, oder externe Ereignisse, wie beispielsweise das Eröffnungsdatum eines Schulsemesters, 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 des ROI 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 mit den Kosten, wenn keine Maßnahmen ergriffen werden.

Technische Anforderungen

Während der Phase der technischen Anforderungen des Lebenszyklus einer Lösung führen Sie eine Anwendungsanalyse durch, bestimmen Anwendungsfälle und ermitteln die Dienstqualitätsanforderungen für die geplante Bereitstellungslösung.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Informationen über technische Anforderungen“ auf Seite 39
- „Anwendungsanalyse“ auf Seite 40
- „Anwendungsfälle“ auf Seite 42
- „Dienstqualitätsanforderungen“ auf Seite 43
- „Anforderungen an die Dienstebene“ auf Seite 53

Informationen über technische Anforderungen

Die Analyse der technischen Anforderungen beginnt mit Geschäftsanforderungsdokumenten, die während der Geschäftsanalysephase des Lebenszyklus einer Lösung erstellt werden. Verwenden Sie die Geschäftsanalyse als Grundlage und gehen Sie wie folgt vor:

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

Die Dienstqualitätsanforderungen werden von der Anwendungsanalyse abgeleitet, wobei die Anwendungsfälle, unter Berücksichtigung der Geschäftsanforderungen und -einschränkungen zuvor ermittelt werden.

Die Geschäftsanforderungen werden später mit logischen Architekturen in der Phase des logischen Konzepts gepaart, um ein Bereitstellungsszenario zu entwerfen. Das Bereitstellungsszenario ist die wichtigste Eingabe für die Bereitstellungskonzeptphase des Lebenszyklus der Lösung.

Wie im Fall der Geschäftsanalyse gibt es keine einfache Formel für die Analyse der technischen Anforderungen, mit der die Anwendungsanalyse, Anwendungsfälle und Systemanforderungen generiert werden. Für die Analyse der technischen Anforderungen müssen die Geschäftsdomäne, die geschäftlichen Ziele und die zugrunde liegende Systemtechnologie begriffen werden.

Anwendungsanalyse

Bei der Anwendungsanalyse werden die verschiedenen Benutzer der zu entwerfenden Lösung sowie die Anwendungsmuster für jene Benutzer ermittelt. Die zusammengetragenen Informationen bieten eine Grundlage für die Einschätzung der Ladebedingungen auf dem System. Die Anwendungsanalyseinformationen sind auch bei der Gewichtung der Anwendungsfälle, wie unter „Anwendungsfälle“ auf Seite 42 beschrieben, von Nutzen.

Während der Anwendungsanalyse sollten Sie Benutzer wannimmer möglich befragen, vorhandene Daten zu Anwendungsmustern untersuchen und Befragungen bei Herstellern und Administratoren von früheren Systemen durchführen. In der folgenden Tabelle werden die Faktoren aufgeführt, die bei der Durchführung einer Anwendungsanalyse berücksichtigt werden sollten.

Tabelle 3-1 Faktoren für die Anwendungsanalyse

Thema	Beschreibung
Anzahl und Art der Benutzer	<p>Stellen Sie fest, wie viele Benutzer Ihre Lösung unterstützen muss, und kategorisieren Sie diese Benutzer gegebenenfalls.</p> <p>Beispielsweise:</p> <ul style="list-style-type: none"> • Eine Business to Customer-(B2C-)Lösung weist möglicherweise eine große Anzahl an Besuchern auf, von denen jedoch nur eine kleine Anzahl bei geschäftlichen Transaktionen registriert und damit befasst ist. • Eine Business to Employee-(B2E-)Lösung enthält normalerweise jeden Mitarbeiter, auch wenn einige Mitarbeiter möglicherweise einen Zugriff von außerhalb des Unternehmensnetzwerks benötigen. <p>In einer B2E-Lösung benötigen Führungskräfte möglicherweise die Berechtigung für Bereiche, auf die der normale Mitarbeiter nicht zugreifen darf.</p>
Aktive und inaktive Benutzer	<p>Ermitteln Sie die Anwendungsmuster und -verhältnisse der aktiven und inaktiven Benutzer.</p> <p>Bei aktiven Benutzern handelt es sich um die Benutzer, die beim System angemeldet sind und mit den Systemdiensten interagieren. Inaktive Benutzer können Benutzer sein, die nicht angemeldet sind; die angemeldet sind, aber nicht mit den Systemkomponenten interagieren, oder Benutzer, die in einer Datenbank gespeichert sind, sich aber nie anmelden.</p>
Administrative Benutzer	<p>Ermitteln Sie Benutzer, die auf das bereitgestellte System zugreifen, um die Bereitstellung zu überwachen, zu aktualisieren und zu unterstützen.</p> <p>Legen Sie sämtliche speziellen administrativen Anwendungsmuster fest, die sich auf die technischen Anforderungen auswirken können (z. B. Verwaltung der Bereitstellung von außerhalb der Firewall aus).</p>
Anwendungsmuster	<p>Bestimmen Sie, in welcher Weise verschiedene Typen von Benutzern auf das System zugreifen und legen Sie Ziele für die erwartete Anwendung fest.</p> <p>Beispielsweise:</p> <ul style="list-style-type: none"> • Gibt es Zeiten, zu denen die Auslastung besonders hoch ist? • Wie sind die normalen Geschäftszeiten? • Sind die Benutzer weltweit verstreut? • Wie lange dauern Benutzerverbindungen erwartungsgemäß?

Tabelle 3-1 Faktoren für die Anwendungsanalyse (*Fortsetzung*)

Thema	Beschreibung
Benutzerwachstum	<p>Stellen Sie fest, ob die Größe des Benutzerstamms festgelegt ist, oder ob die Bereitstellung einen Anstieg der Anzahl der Benutzer zu erwarten hat.</p> <p>Wenn der Benutzerstamm wahrscheinlich wächst, versuchen Sie vernünftige Prognosen hinsichtlich des Wachstums zu machen.</p>
Benutzertransaktionen	<p>Bestimmen Sie die Typen der Benutzertransaktionen, die unterstützt werden müssen. Diese Benutzertransaktionen können in Anwendungsfälle übertragen werden.</p> <p>Beispielsweise:</p> <ul style="list-style-type: none"> • Welche Aufgaben führen die Benutzer durch? • Bleiben Benutzer nach ihrer Anmeldung angemeldet? Führen sie normalerweise einige Aufgaben durch und melden sich wieder ab? • Sind für eine sinnvolle Zusammenarbeit zwischen den Benutzern gemeinsame Kalender, Internetkonferenzen und interne Webseiten erforderlich?
Benutzerstudien und statistische Daten	<p>Verwenden Sie bereits vorhandene Benutzerstudien und andere Quellen, um Muster für das Benutzerverhalten zu ermitteln.</p> <p>Oft liegen in Firmen oder Industrieunternehmen Benutzerstudien vor, denen Sie nützliche Informationen über die Benutzer entnehmen können. Protokolldateien für vorhandene Anwendungen können statistische Daten enthalten, die bei der Einschätzung für ein System nützlich sind.</p>

Anwendungsfälle

Anwendungsfälle stellen die typische Benutzerinteraktion mit der zu entwerfenden Lösung beispielhaft dar und beschreiben den vollständigen Verlauf eines Vorgangs aus der Perspektive eines Endbenutzers. Wenn für das Konzept eine komplette Gruppe von Anwendungsfällen zugrunde gelegt wird, wird das Augenmerk dauerhaft darauf gelegt, dass die gewünschte Funktionalität erbracht wird. Anwendungsfälle sind die wichtigste Grundlage für ein logisches Konzept.

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 durch zwei Ebenen näher beschrieben werden.

- **Anwendungsfallberichte.** Beschreibungen der einzelnen Anwendungsfälle, einschließlich der primären und alternativen Ereignisabläufe.
- **Anwendungsfalldiagramme.** Diagramme, die die Beziehungen zwischen den Beteiligten und die Anwendungsfälle bildlich darstellen und dadurch eine formellere Organisation des Ereignisflusses bieten.
Anwendungsfalldiagramme eignen sich zur Darstellung langer oder komplexer Anwendungsfälle. Normalerweise verwenden Sie Unified Modeling Language-(UML-)Standards, um Anwendungsfalldiagramme zu zeichnen.

Dienstqualitätsanforderungen

Dienstqualitätsanforderungen (QoS-Anforderungen) sind technische Spezifikationen, die die Dienstqualität von Merkmalen, wie beispielsweise der Leistung, Verfügbarkeit, Skalierbarkeit und Zweckmäßigkeit, bestimmen. QoS-Anforderungen werden durch die geschäftlichen Bedürfnisse bestimmt, die durch die Anforderungen für das Unternehmen angegeben werden. Wenn beispielsweise ein Dienst 24 Stunden jeden Tag im gesamten Jahr verfügbar sein muss, muss die Verfügbarkeitsanforderung auf diese Anforderung im Unternehmen abzielen.

In der folgenden Tabelle werden die Systemqualitäten aufgeführt, die normalerweise die Grundlage für die QoS-Anforderungen bilden.

Tabelle 3-2 Systemqualitäten mit Auswirkung auf dieQoS-Anforderungen

Systemqualität	Beschreibung
Leistung	Die Messung der Antwortzeit und des Datendurchsatzes in Bezug auf die Benutzerladebedingungen.
Verfügbarkeit	Ein Maß für die Häufigkeit, mit der die Ressourcen und Dienste eines Systems für Endbenutzer verfügbar sind, wird oft als <i>Betriebszeit</i> eines Systems bezeichnet.
Skalierbarkeit	Die Fähigkeit, einem bereitgestellten System im Laufe der Zeit Kapazität (und Benutzer) hinzuzufügen. Die Skalierbarkeit umfasst in der Regel das Hinzufügen von Ressourcen zum System, sollte jedoch keine Änderungen an der Bereitstellungsarchitektur erfordern.
Sicherheit	Eine komplexe Kombination von Faktoren, die die Integrität eines Systems und seiner Benutzer beschreibt. Zur Sicherheit zählt auch die Authentifizierung und Autorisierung von Benutzern, die Datensicherheit und der sichere Zugriff auf ein bereitgestelltes System.
Latente Kapazität	Die Fähigkeit eines Systems, eine außergewöhnliche Spitzenauslastung ohne zusätzliche Ressourcen zu bewältigen. Die latente Kapazität ist ein Faktor bei den Qualitäten Verfügbarkeit, Leistung und Skalierbarkeit.
Zweckmäßigkeit	Die Einfachheit der Wartung eines bereitgestellten Systems, einschließlich der Überwachung des Systems, der Behebung von auftretenden Problemen und der Aufrüstung der Hardware- und Softwarekomponenten.

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 weiterer Server, um Verfügbarkeitsprobleme zu beheben, wirkt sich auf die Wartungskosten (Zweckmäßigkeit) aus.

Für die Entwicklung eines Systems, das sowohl die geschäftlichen Anforderungen als auch Einschränkungen befriedigt, muss klar sein, wie die Systemqualitäten verknüpft sind und wie diese abgestimmt werden.

In den folgenden Abschnitten werden die Systemqualitäten, die sich auf das Bereitstellungs-konzept auswirken, näher beschrieben. Es wird außerdem erklärt, welche Faktoren bei der Gestaltung der QoS-Anforderungen berücksichtigt werden müssen. Ein Abschnitt zu den Anforderungen für 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 normalerweise nur bis zu vier Sekunden.

Untersuchen Sie, beginnend mit dieser Geschäftsanforderung sämtliche Anwendungsfälle, um zu bestimmen, wie Sie diese Anforderung auf einer Systemebene ausdrücken können. In einigen Fällen möchten Sie möglicherweise Benutzerladebedingungen miteinschließen, die während der Anwendungsanalyse ermittelt wurden. Drücken Sie die Leistungsanforderung für jeden Anwendungsfall in Bezug auf die Antwortzeit unter bestimmten Ladebedingungen oder in Bezug auf die Antwortzeit zuzüglich Datendurchsatz aus. Geben Sie auch die zulässige Anzahl an Fehlern an.

Im Folgenden sehen Sie zwei Beispiele für die Angabe der Systemanforderungen für die Leistung:

Die Antwortzeit bei der Webseitenaktualisierung darf während des gesamten Tages 4 Sekunden nicht überschreiten (Prüfung alle 15 Minuten), wobei nur unter 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 außergewöhnlichen Spitzenauslastung) verknüpft.

Verfügbarkeit

Über die Verfügbarkeit kann man die Betriebszeit eines Systems bestimmen. Sie bemisst sich normalerweise aus dem Prozentsatz der Zeit, während der Benutzer auf das System zugreifen kann. 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 Aufrüstungen) heruntergefahren wird, werden nicht als Ausfallzeit bezeichnet. Ein grundlegende Gleichung zur Berechnung der Systemverfügbarkeit als Prozentsatz der Betriebszeit ist:

$$\text{Verfügbarkeit} = \text{Betriebszeit} / (\text{Betriebszeit} + \text{Ausfallzeit}) * 100\%$$

Normalerweise messen Sie die Verfügbarkeit anhand der Anzahl der „Neunen“, die Sie erreichen können. Beispielsweise erhalten Sie bei 99%-iger Verfügbarkeit zwei Neunen. Die Angabe zusätzlicher Neunen wirkt sich erheblich auf das Bereitstellungskonzept aus. In der folgenden Tabelle wird die unvorhergesehene Ausfallzeit bei zusätzlichen „Verfügbarkeits-Neunen“ auf einem System berechnet, das 24 Stunden an 7 Tagen die Woche im gesamten Jahr (insgesamt 8760 Stunden) verfügbar ist.

Tabelle 3-3 Unvorhergesehene Ausfallzeiten für ein System, das das ganze Jahr (8760 Stunden) läuft

Anzahl der Neunen	Verfügbarkeit (Prozent)	Unvorhergesehene Ausfallzeiten
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 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 Einzelpunkt-Versagen 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 Einzelpunkt-Versagen ermitteln und vermeiden.

Fehlertolerante Systeme können in der Implementierung und Wartung teuer sein. Stellen Sie sicher, dass Sie die 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 Anwendungsfälle und Anwendungsanalysen verweisen, für die eine höhere Verfügbarkeit erforderlich ist.

Es ist hilfreich, Verfügbarkeitsbedürfnisse anhand von verschiedenen sortierten Prioritäten aufzulisten. In der folgenden Tabelle wird die Verfügbarkeit verschiedener Diensttypen 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 erforderlich erachtet werden.

Dienstausfall

Im Rahmen des Verfügbarkeitskonzepts wird auch berücksichtigt, was geschieht, 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.

Sie geben die Skalierbarkeitsanforderungen nicht notwendigerweise mit den QoS-Anforderungen an, es sei denn, das geplante Wachstum der Bereitstellung wird eindeutig in den Geschäftsanforderungen festgehalten. Während der Bereitstellungsphase des Lösungszyklus 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 eingeschlossen 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 Branchendaten oder Markteinschätzungen.
Konzept in Bezug auf maximal sinnvollen Umfang	Konzepterstellung mit Hinblick auf den maximal erforderlichen Umfang sowohl für den bekannten als auch für den möglichen Bedarf. Häufig kann es sich hierbei um eine Schätzung über 24 Monate handeln, der die Leistungseinschätzung der vorhandenen Benutzerauslastung und angemessene Einschätzungen für die zukünftige Auslastung zugrunde liegen. Der Zeitraum für die Schätzung hängt in hohem Maße von der Zuverlässigkeit der Prognosen ab.

Tabelle 3-5 Skalierbarkeitsfaktoren (*Fortsetzung*)

Thema	Beschreibung
Festlegen angemessener Meilensteine	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. Beispielsweise: <ul style="list-style-type: none"> • Kapitalbeschaffung (z. B. vierteljährlich oder jährlich) • Vorlaufzeit für den Erwerb von Hardware oder Software (z. B. eine bis sechs Wochen) • Puffer (10 % bis 100 % je nach Wachstumserwartungen)
Aufnahme neuer Technologien	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.

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 die Entwicklung und Implementierung der Sicherheitsmaßnahmen leiten soll.

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, Unernehmensdaten 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

Aspekt	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>

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

Aspekt	Beschreibung
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 geographisch 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>
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 vonnöten ist. 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 55
- „Entwurf einer logischen Architektur“ auf Seite 57
- „Beispiele für logische Architekturen“ auf Seite 65
- „Zugriffszonen“ auf Seite 75
- „Bereitstellungsszenario“ auf Seite 76

Informationen zu logischen Architekturen

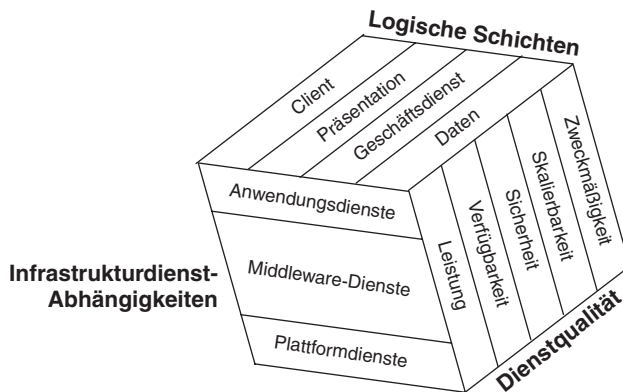
In einer logischen Architektur werden die Softwarekomponenten identifiziert, die für die Implementierung einer Lösung erforderlich sind, und die gegenseitigen Abhängigkeiten der Komponenten werden dargestellt. Die logische Architektur und die Dienstqualitätsanforderungen, die während der Phase der technischen 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 sind zudem in [Abbildung 4-1 auf Seite 56](#) dargestellt.

- **Infrastrukturdienst-Abhängigkeiten.** Interagierende Softwarekomponenten, die von Enterprise Services 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 die Leistung, Verfügbarkeit, Skalierbarkeit und sonstige Eigenschaften, die bestimmte Aspekte des Konzepts und der Funktion einer Softwarelösung darstellen.

Abbildung 4-1 Die drei Pfeiler der Java Enterprise System-Lösungsarchitektur



HINWEIS Weitere Informationen zu den Konzepten der Java Enterprise System-Architektur erhalten Sie im Kapitel zur Java Enterprise System-Architektur im *Java Enterprise System Technical Overview*, <http://docs.sun.com/doc/819-0061>.

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.

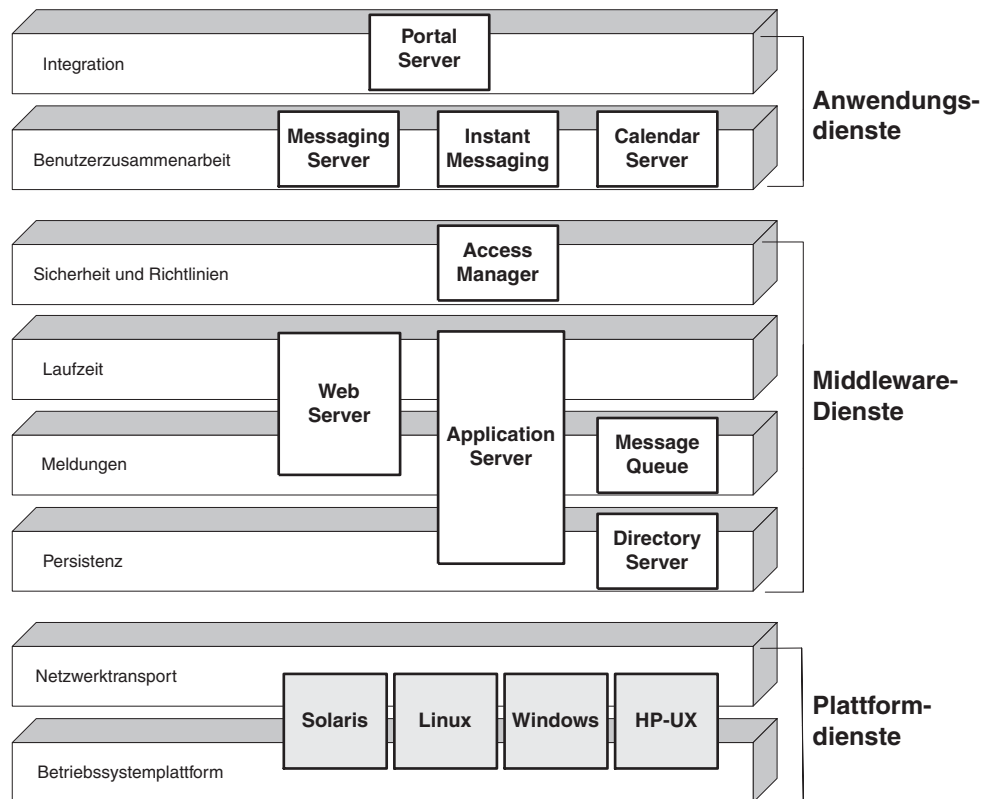
Sie platzieren die Java Enterprise System-Komponenten entsprechend der von Ihnen bereitgestellten Art von Diensten innerhalb des Kontexts einer mehrschichtigen Architektur. Das Verständnis der Komponenten als Bestandteil einer mehrschichtigen Architektur hilft Ihnen später dabei, die Verteilung der von 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 75 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. Der *Java Enterprise System Technical Overview*, <http://docs.sun.com/doc/819-0061>, bietet zusätzliche Informationen zu den Java Enterprise System-Komponenten und zu den von ihnen bereitgestellten Diensten.

Abbildung 4-2 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 angeben. Wenn Sie beispielsweise Messaging Server als erforderliche Komponente in einer logischen Architektur bestimmen, muss die logische Architektur ebenfalls Directory Server und möglicherweise auch 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 der Java Enterprise System-Komponenten aufgelistet. Eine bildliche Darstellung der Abhängigkeiten zwischen den Schlüsselkomponenten finden Sie in [Abbildung 4-3 auf Seite 61](#). Verwenden Sie beim Entwurf einer logischen Architektur folgende Tabelle und die dazugehörige Abbildung, um abhängige Komponenten in Ihrer Struktur zu bestimmen.

Tabelle 4-1 Java Enterprise System Komponentenabhängigkeiten

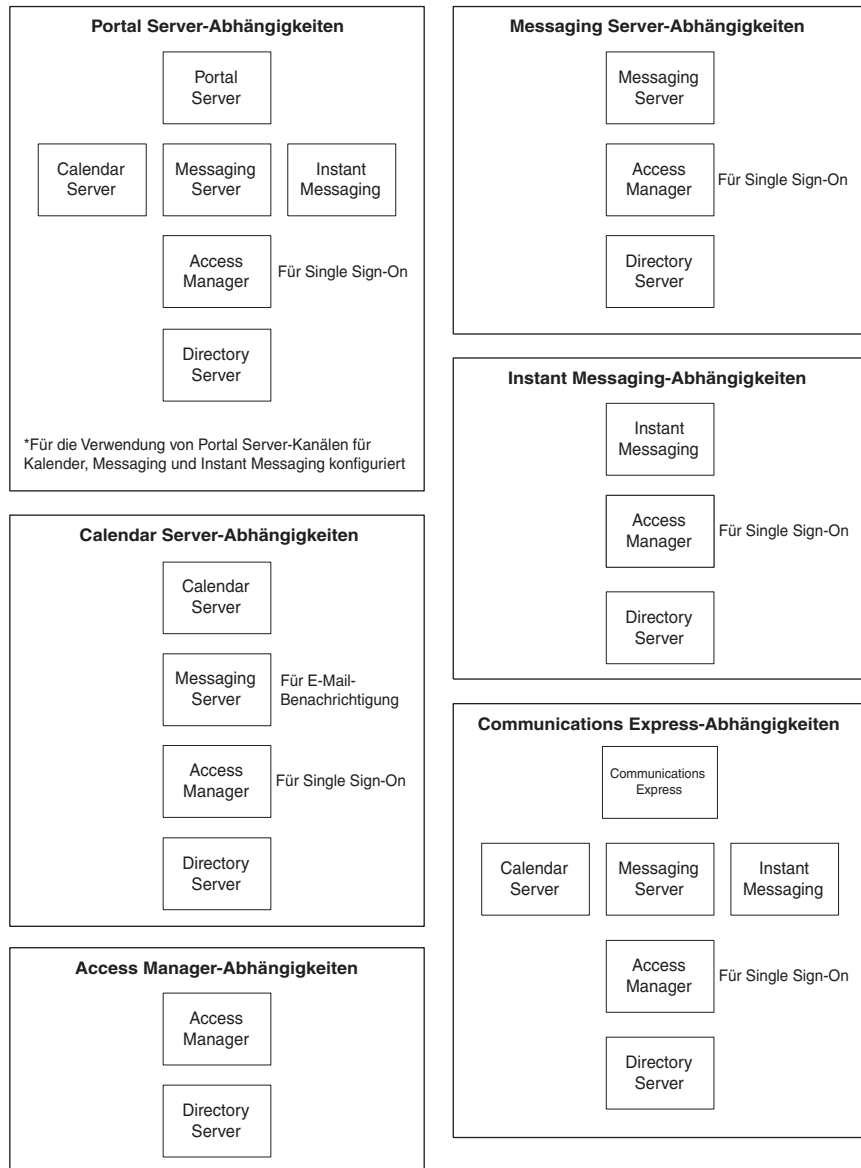
Java Enterprise System-Komponente	Ist abhängig von
Application Server	Message Queue Directory Server (optional)
Calendar Server	Messaging Server (für den E-Mail-Benachrichtigungsdienst) Access Manager (für Single Sign-On) Web Server (für die Webschnittstelle) Directory Server
Communications Express	Access Manager (für Single Sign-On) Calendar Server Messaging Server Instant Messaging Web Server (für die Webschnittstelle) Directory Server
Directory Proxy Server	Directory Server
Directory Server	Keine
Access Manager	Application Server oder Web Server Directory 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

Tabelle 4-1 Java Enterprise System Komponentenabhängigkeiten (*Fortsetzung*)

Java Enterprise System-Komponente	Ist abhängig von
Portal Server	Wenn zur Verwendung von Portal Server-Kanälen konfiguriert: Calendar Server Messaging Server Instant Messaging Access Manager (für Single Sign-On) Application Server oder Web Server Directory Server
Portal Server Secure Remote Access	Portal Server
Web Server	Access Manager (optional, für Zugriffssteuerung)

HINWEIS Die Abhängigkeiten zwischen den in [Tabelle 4-1](#) aufgelisteten Java Enterprise System-Komponenten sind nicht vollständig. In [Tabelle 4-1](#) werden keine Abhängigkeiten angegeben, die bei der Planung der Installation zu berücksichtigen sind. Eine vollständige Liste der Java Enterprise System-Abhängigkeiten finden Sie im *Java Enterprise System-Installationshandbuch*, <http://docs.sun.com/doc/819-0805>.

Abbildung 4-3 Java Enterprise System-Komponentenabhängigkeiten



Webcontainer-Unterstützung

Im vorigen Abschnitt „[Komponentenabhängigkeiten](#)“ 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 logisch eindeutigen Dienste konfiguriert werden:

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

Diese unterschiedlichen Konfigurationen von Messaging Server bieten Funktionen, die auf separaten physischen Servern bereitgestellt werden und in unterschiedlichen 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 „[Beispiele für logische Architekturen](#)“ auf Seite 65 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 einigen Konfigurationen von Messaging Server kann darüber hinaus der 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

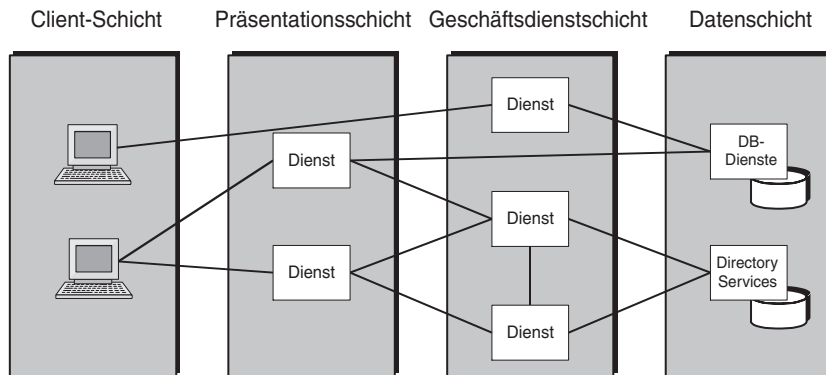
Komponente	Beschreibung
Directory Proxy Server	Bietet eine verbesserte Steuerung des Verzeichniszugriffs, der Schemakompatibilität, der Weiterleitung und Lastenverteilung für mehrere Directory Server-Instanzen.
Portal Server, Portal Server Secure Remote Access	Ermöglicht außerhalb einer Unternehmens-Firewall 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 ServerMessage 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 75](#) 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 4-4 Mehrschichtiges Architekturmodell



In der folgenden Tabelle werden die in [Abbildung 4-4](#) 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.

Tabelle 4-4 Logische Schichten in einer mehrschichtigen Architektur (*Fortsetzung*)

Schicht	Beschreibung
Geschäftsdienstschicht	Bietet Back-End-Dienste, mit denen in der Regel Daten aus der Datenschicht abgerufen werden, um Dienste innerhalb der Präsentations- oder Geschäftsdienstschicht direkt für die Clients in der Client-Schicht abzurufen. 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 Geschäftsdienstschicht 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 leichter 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.

Beispiele 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.

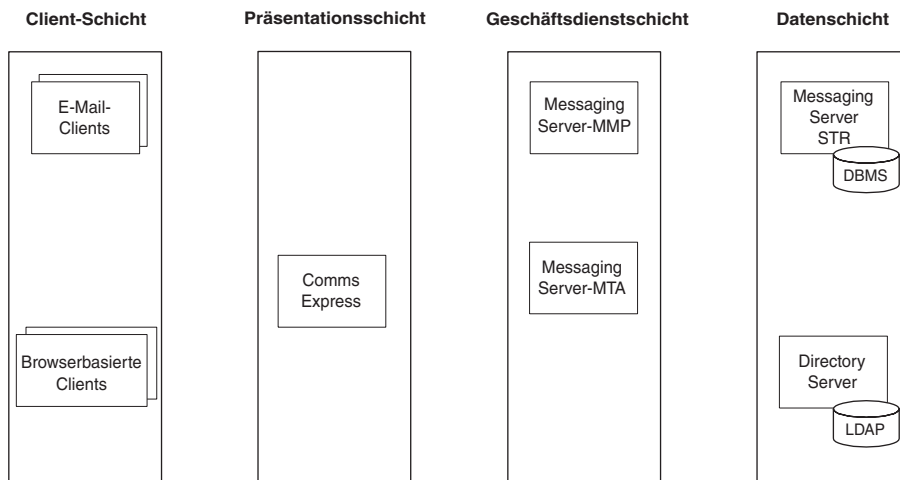
Beim ersten Beispiel handelt es sich um eine grundlegende Messaging Server-Lösung, die verdeutlicht, wie logisch eindeutige Komponenten von Messaging Server mit anderen Komponenten interagieren. Im zweiten Beispiel wird eine logische Architektur für eine identitätsbasierte Bereitstellungslösung dargestellt, 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 71 beschrieben.

Abbildung 4-5 Logische Architektur für die Messaging Server-Bereitstellung



In der folgenden Tabelle werden die in [Abbildung 4-5](#) 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 [Abbildung 4-5 auf Seite 66](#) 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

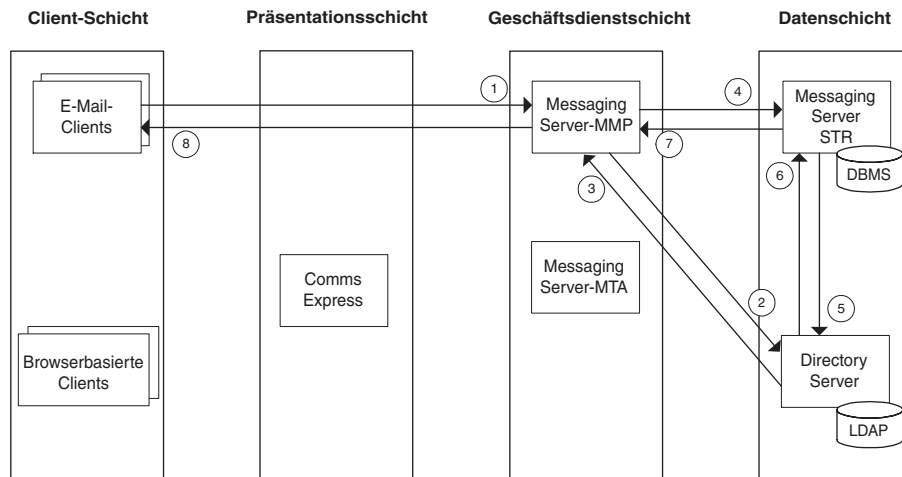
Mithilfe von Anwendungsfällen können Sie Beziehungen zwischen den logischen Komponenten einer Architektur bestimmen. Durch die Zuordnung der Interaktionen zwischen den Komponenten entsprechend den Anwendungsfällen erhalten Sie eine visuelle Darstellung der Komponenteninteraktion, die hilfreich für die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

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.

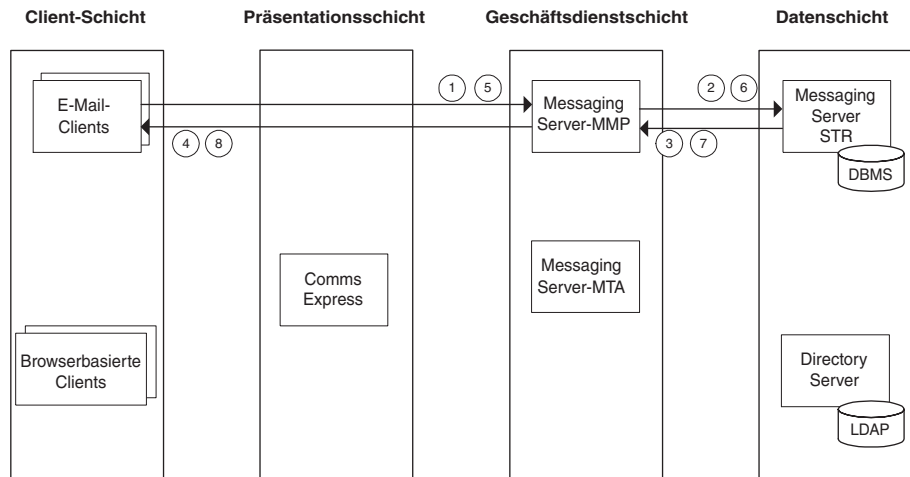
Abbildung 4-6 Logische Architektur von Messaging Server, in der Anwendungsfall 1 dargestellt wird



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

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.

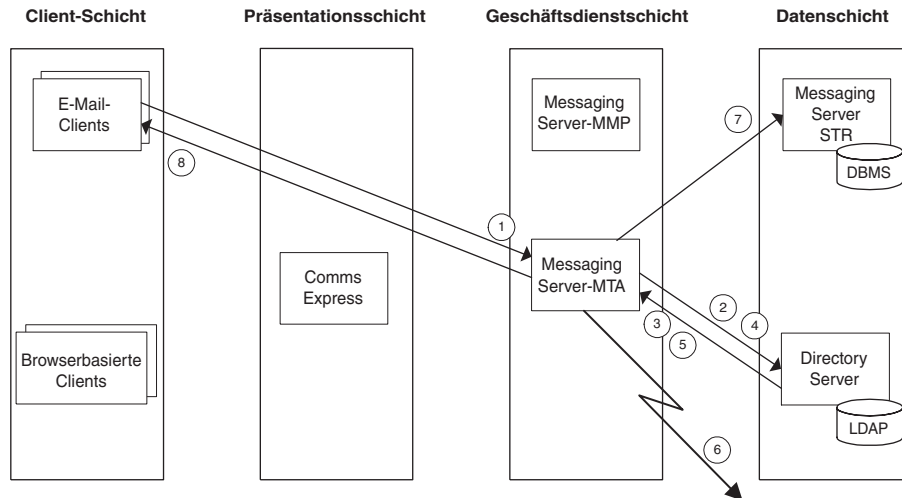
Abbildung 4-7 Logische Architektur von Messaging Server, in der Anwendungsfall 2 dargestellt wird



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

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 leitet 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.

Abbildung 4-8 Logische Architektur von Messaging Server, in der Anwendungsfall 3 dargestellt wird



Identitätsbasiertes Kommunikationsbeispiel

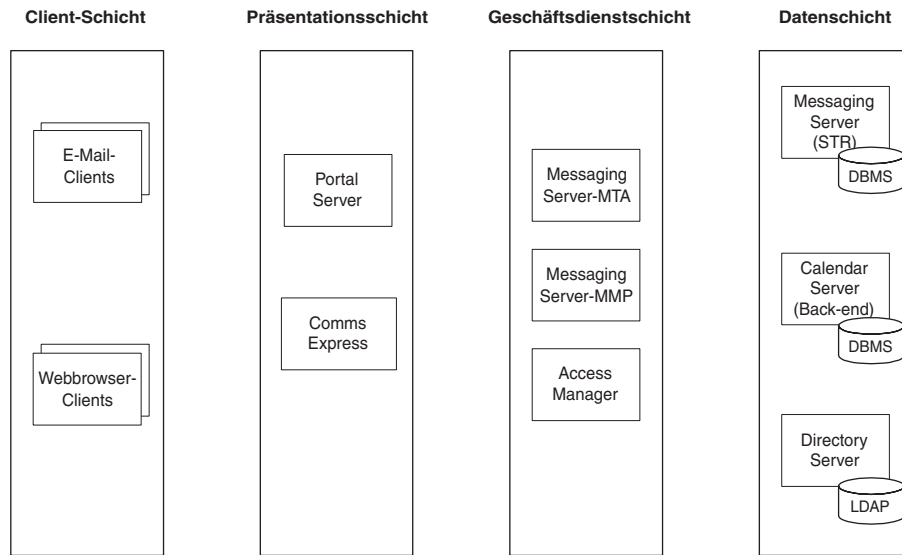
In diesem Beispiel wird eine identitätsbasierte Kommunikationslösung für ein Unternehmen mittlerer Größe mit etwa 1000 bis 5000 Mitarbeitern dargestellt. In der Regel ist eine ausführliche Geschäftsanalyse mit anschließender detaillierter Bestimmung der technischen Anforderungen erforderlich, um die 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 Sign-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.

Abbildung 4-9 Logische Architektur für identitätsbasiertes Kommunikationsszenario



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 Anmeldungsszenario in zwei Anwendungsfälle unterteilt:

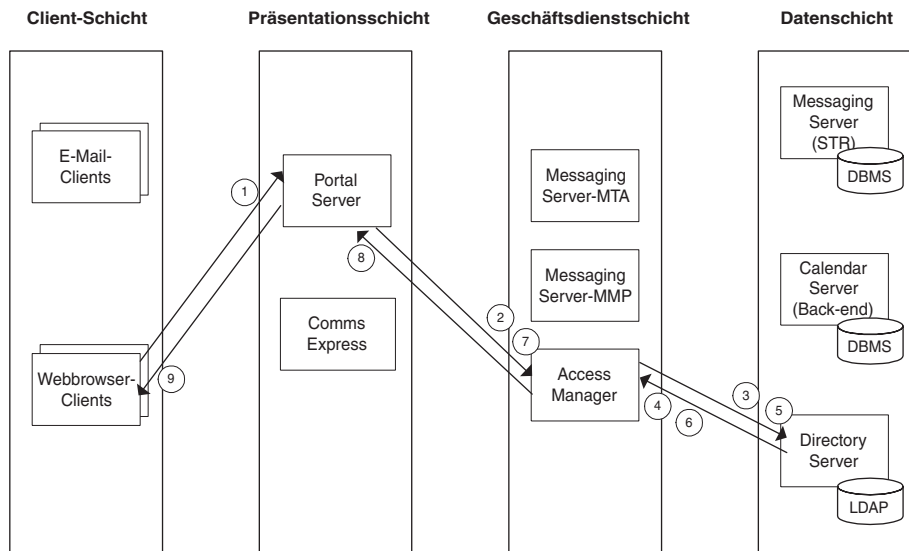
- Der Benutzer meldet sich an, wird authentifiziert und Portal Server ruft die Portalkonfiguration des Benutzers ab.
- Portal Server ruft E-Mail- und Kalenderinformationen zur Anzeige im Web-Client ab.

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

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.

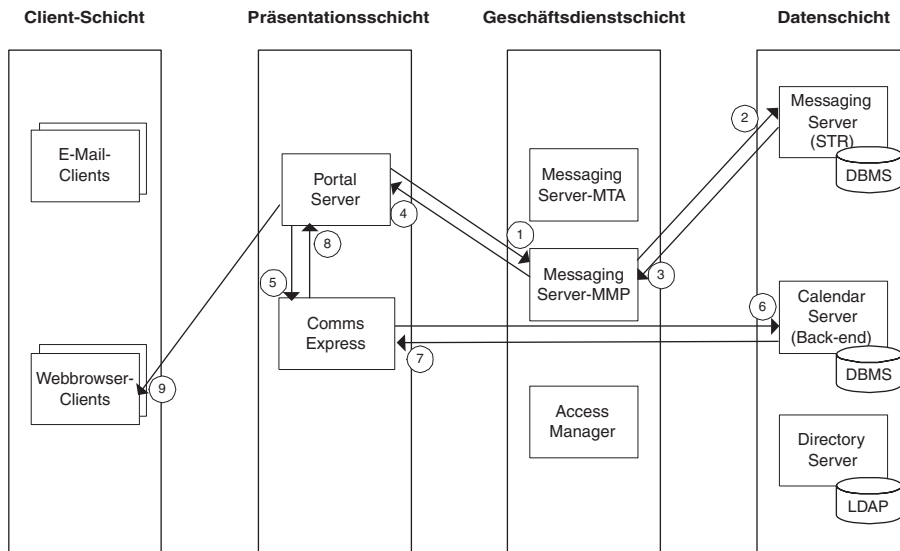
Abbildung 4-10 Logische Architektur des Kommunikationsszenarios, in der Anwendungsfall 1 dargestellt wird



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

1. Nach erfolgter Anmeldung, Authentifizierung und dem Abrufen 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.

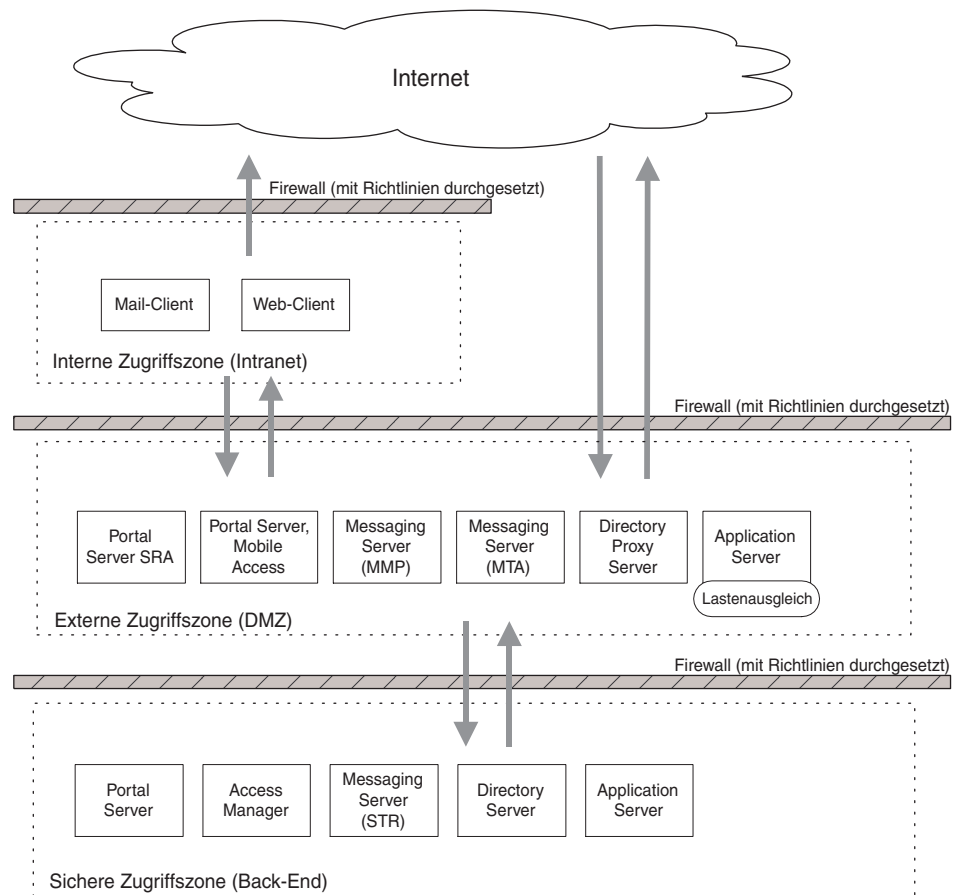
Abbildung 4-11 Logische Architektur des Kommunikationsszenarios, in der Anwendungsfall 2 dargestellt wird



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 4-12 In Zugriffszonen platzierte logische Komponenten



In der folgenden Tabelle werden die in [Abbildung 4-12](#) dargestellten Zugriffszonen beschrieben.

Tabelle 4-6 Sichere Zugriffszonen und darin platzierte Komponenten

Zugriffszone	Beschreibung
Interne Zugriffszone (Intranet)	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. 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.
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 [Abbildung 4-12](#) 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 Bereitstellungsszenario dar. Das Bereitstellungsszenario ist der Ausgangspunkt für den Entwurf der Bereitstellungsarchitektur, wie in [Kapitel 5](#), „[Bereitstellungskonzept](#)“. 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:

- „Bereitstellungskonzept“ auf Seite 78
- „Bereitstellungskonzept Methodik“ auf Seite 82
- „Einschätzen von Prozessoranforderungen“ auf Seite 83
- „Einschätzen von Prozessoranforderungen für sichere Transaktionen“ auf Seite 90
- „Ermitteln von Verfügbarkeitsstrategien“ auf Seite 94
- „Ausarbeiten von Strategien hinsichtlich der Skalierbarkeit“ auf Seite 103
- „Konzeption für optimale Ressourcennutzung“ auf Seite 109
- „Beispiel für eine Bereitstellungsarchitektur“ auf Seite 112

Bereitstellungskonzept

Die Erstellung eines Bereitstellungskonzepts beginnt mit dem Bereitstellungsszenario, das während der Phasen des logischen Designs und der technischen Anforderungen des Lebenszyklus der Lösung erstellt wurde. Das Bereitstellungsszenario umfasst eine logische Architektur und die für die Lösung erforderlichen Anforderungen hinsichtlich der Dienstqualität (Quality of Service, QoS). Die in der logischen Architektur identifizierten Komponenten werden für physische Server und andere Netzwerkgeräte übergreifend zugeordnet, um eine Bereitstellungsarchitektur zu erstellen. Die QoS-Anforderungen bieten Anhaltspunkte hinsichtlich der Hardwarekonfiguration in Bezug auf Leistung, Verfügbarkeit, Skalierbarkeit und andere QoS-Spezifikationen in diesem Zusammenhang.

Bei der Konzeption der Bereitstellungsarchitektur handelt es sich um einen von Wiederholungen geprägten Vorgang. Im Normalfall werden die QoS-Anforderungen erneut zurate gezogen und die im Vorfeld festgelegten Konzepte nochmals überprüft. Hierbei wird die Wechselbeziehung der QoS-Anforderungen in Betracht gezogen 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 im Rahmen der Bereitstellungskonzeptphase, im Normalfall nach der Erstellung der Bereitstellungsarchitektur. Die geschätzten realen Bereitstellungskosten werden anhand der Bereitstellungsarchitektur und ggf. 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 Bereitstellungskonzeptphase 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.** Hier werden die Strategien und Vorgänge zur Migration von Unternehmensdaten und zur Aufrüstung von Unternehmenssoftware erläutert. Die migrierten Daten müssen den Formaten und Standards der neu installierten Unternehmensanwendungen entsprechen. Voraussetzung für die Interoperabilität ist, dass von jeglicher 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, bei denen das in der Bereitstellungsarchitektur angegebene Kontoreplikationskonzept beachtet wird, sowie Vorgehensweisen zur Bereitstellung neuer Inhalte für Verzeichnisse.

- **Testplan.** Hier wird beschrieben, wie die bereitgestellte Software getestet wird, einschließlich spezifischer Pläne für die Bereitstellung von Prototyp- und Pilotimplementierungen. Ein weiterer Bestandteil sind Belastungstests, bei denen die Fähigkeit zur Handhabung der geplanten Auslastung ermittelt wird; bei Funktionstests wird überprüft, ob der Betrieb erwartungsgemäß verläuft.
- **Roll-out-Plan.** Hier werden die Vorgehensweisen sowie der Terminplan für die Implementierung aus einer Plan- und Testumgebung in eine Produktionsumgebung erläutert. 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.** Hier wird erläutert, 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 dem Überwachungs-, Wartungs-, Installations- und Aufrüstungsaspekte angesprochen werden.
- **Schulungsbuch.** Enthält Vorgänge und Vorgehensweisen zur Schulung der Personen, die mit der neu installierten Software arbeiten werden (u. a. Administratoren und Endbenutzer).

Faktoren mit Auswirkung auf das Bereitstellungskonzept

Die Entscheidungen, die Sie in der Bereitstellungskonzeptphase treffen, werden von mehreren Faktoren beeinflusst. Ziehen Sie folgende Hauptfaktoren in Betracht:

- **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 hinsichtlich der Dienstqualität (Quality of Service, QoS) werden unterschiedliche Aspekte des Betriebs einer Lösung angegeben. Anhand der QoS-Anforderungen können Sie Strategien ausarbeiten, 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 auf die technischen Anforderungen bezogenen Phase im Lösungszyklus erstellt wird, 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 auf die technischen Anforderungen bezogenen Phase im Lösungszyklus 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. Wenn diese Anforderungen nicht erfüllt werden, geht hieraus ebenfalls hervor, auf welcher Ebene und in welchem Umfang Kundenunterstützung vonnöten ist. Die in einer Vereinbarung auf Dienstebene angegebenen Leistungsanforderungen sollten von einem Bereitstellungskonzept problemlos erfüllt werden.

- **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 beachtet 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.

Bereitstellungskonzept Methodik

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 dreht es sich im Normalfall hauptsächlich darum, Leistungsvorgaben zu erfüllen und gleichzeitig anderen QoS-Anforderungen gerecht zu werden. Mit den von Ihnen eingesetzten Strategien muss ein Gleichgewicht der Tradeoffs Ihres Konzepts 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 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 Proessoreinschätzungen 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 stellt sich beispielsweise folgende Frage: Welche Auswirkungen hat die Verfügbarkeits- und Skalierbarkeitsstrategie auf die Wartungseignung des Systems? Welche sonstigen Kosten entstehen durch die Strategien?
- **Identifizierung 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. Hier wird anhand eines Beispielszenarios einer Kommunikationsbereitstellung 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 zur Betrachtung der Faktoren dar, die sich auf das Endergebnis auswirken.

1. Ermitteln einer CPU-Basiseinschätzung für Komponenten, die für das System als Benutzereinstiegsunkte 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.

Sehen Sie sich die Interaktion zwischen Komponenten in der logischen Architektur genau an, 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 Bereitstellungskonzept 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 71 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

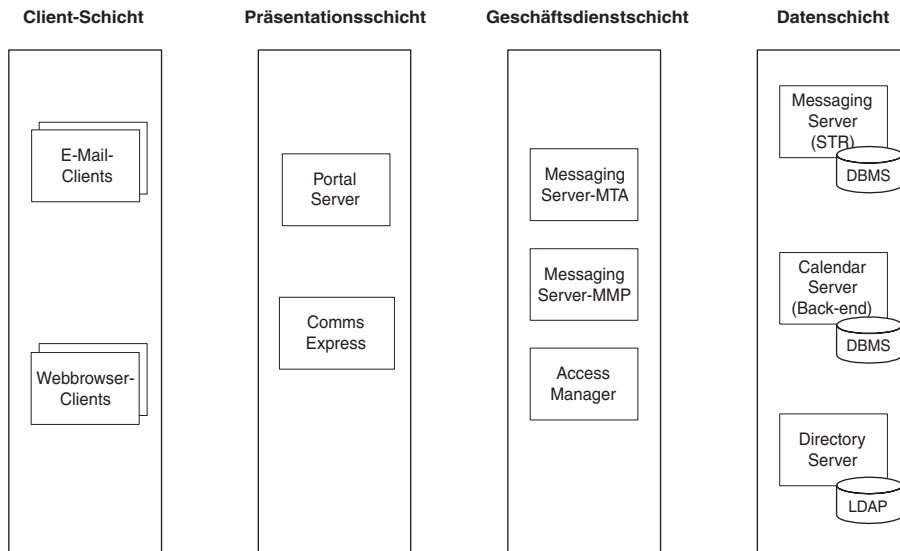
- Spezifische Kosten und Spezifikationen von Verarbeitungs- und Netzwerkhardware
- Bei der Implementierung vergleichbarer Bereitstellungen gesammelte Erfahrungen

ACHTUNG Die in diesen Beispielen enthaltenen Informationen sollten nicht als spezifische Implementierungshinweise angesehen werden, sie dienen lediglich der Veranschaulichung eines Vorgangs, den Sie bei der Konzeption eines Systems möglicherweise durchführen.

Ermitteln der CPU-Basiseinschätzung für Benutzereinstiegspunkte

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. Nachfolgend ist die logische Architektur für ein identitätsbasiertes Kommunikationsszenario dargestellt, das zuvor ([Kapitel 4, „Logisches Konzept“ auf Seite 71](#)) beschrieben wurde.

Abbildung 5-1 Logische Architektur für ein identitätsbasiertes Kommunikationsszenario



In der nachfolgenden 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 Benutzereinstiegspunkt ist.
Communications Express	2	Leitet Daten an Messaging- und Kalender-Kanäle von Portal Server weiter.

CPU-Einschätzungen für Dienstabhängigkeiten aufnehmen

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 von Interaktion zwischen Komponenten genau angegeben werden, wie im Beispiel mit der logischen Architektur im Abschnitt „[Beispiele für logische Architekturen](#)“ auf Seite 65 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 Empfänger weiter.
Messaging Server MMP	1	Ermöglicht den Zugriff auf den 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.

Tabelle 5-2 CPU-Einschätzungen für unterstützende Komponenten (*Fortsetzung*)

Komponente	CPUs	Beschreibung
Directory Server	2	Stellt LDAP-(Lightweight Directory Access Protocol-)Verzeichnisdienste zur Verfügung.
Web Server	0	Bietet Webcontainer-Unterstützung für Portal Server und Access Manager. (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 der auf die technischen Anforderungen bezogenen Phase, wie sich der sichere Transport von Daten auf die Auslastungsanforderungen auswirken kann, und passen Sie Ihre Einschätzungen entsprechend an. Im nachfolgenden Abschnitt, „[Einschätzen von Prozessoranforderungen für sichere Transaktionen](#)“ auf Seite 90, wird erläutert, wie 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 nachfolgenden Abschnitt, „[Ermitteln von Verfügbarkeitsstrategien](#)“ auf Seite 94, werden Größenaspekte für Verfügbarkeitslösungen erläutert. Im Abschnitt „[Ausarbeiten von Strategien hinsichtlich der Skalierbarkeit](#)“ auf Seite 103 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 auf Ihre spezifischen Anforderungen hinsichtlich der Replikation von Diensten eingehen.

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 nachfolgenden 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

Komponente	CPUs	Arbeitsspeicher
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 wie 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 vonnöten. In anderen Szenarios ist der sichere Transport möglicherweise für sämtliche Transaktionen Bedingung.

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 weiterführend erläutert, um aufzuzeigen, wie CPU-Anforderungen für einen theoretischen Anwendungsfall berechnet werden können.

Führen Sie zur Einschätzung der CPU-Anforderungen für sichere Transaktionen 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 85, 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 [Tabelle 5-5](#) finden Sie eine Beispiellberechnung, 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 Tabelle 5-3 auf Seite 88 beträgt 4 CPUs.	-----
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 5fache an Prozessorkapazität erforderlich ist.	Für 10 Prozent der Basiseinschätzung ist der sichere Transport erforderlich: $0,10 \times 4 \text{ CPUs} = 0,4 \text{ CPUs}$ Erhöhen Sie die CPU-Kapazität für sichere Transaktionen um den Faktor 4: $4 \times 0,4 = 1,6 \text{ CPUs}$	1,6 CPUs
3	Berechnen Sie die verringerten CPU-Einschätzungen für nicht sichere Transaktionen.	90 Prozent der Basiseinschätzung sind nicht sicher: $0,9 \times 4 \text{ CPUs} = 3,6 \text{ CPUs}$	3,6 CPUs
4	Berechnen Sie die angepassten CPU-Gesamteinschätzungen für sichere und nicht sichere Transaktionen.	Sichere Einschätzung + nicht sichere Einschätzung = Gesamtwert: $1,6 \text{ CPUs} + 3,6 \text{ CPUs} = 5,2 \text{ CPUs}$	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 [Tabelle 5-5 auf Seite 92](#) dahin gehend ändern, dass zwei zusätzliche CPUs und 4 GB Speicher hinzugefügt und so der Gesamtwert für Portal Server ermittelt wird.

Tabelle 5-6 Anpassungen der CPU-Einschätzung für sichere Portal Server-Transaktionen

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 dem Installieren, Konfigurieren, 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 der 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 (ziehen Sie hierzu die Erläuterung unter [„Ausarbeiten von Strategien hinsichtlich der Skalierbarkeit“](#) auf Seite 103 zurate). Komplexe Lösungen mit großem Verwaltungs- und Wartungsaufwand sollten vermieden werden.

Verfügbarkeitsstrategien

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

- **Lastenausgleich.** Nutzt redundante Hard- und Softwarekomponenten zur verteilten Lastenverarbeitung. Von einem Lastenausgleichssystem werden sämtliche Anforderungen eines Diensts an eine von mehreren symmetrischen Instanzen des Diensts 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.

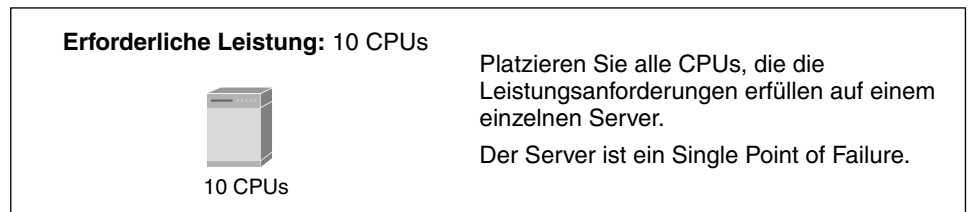
- **Replikation von Diensten.** Durch die Replikation von Diensten werden mehrere Zugriffsmöglichkeiten auf dieselben Daten bereitgestellt. Directory Server enthält zahlreiche Replikations- und Synchronisierungsstrategien für den LDAP-Verzeichniszugriff.

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.

Abbildung 5-2 System mit nur einem Server



Sun bietet High-End-Server mit folgenden Vorteilen:

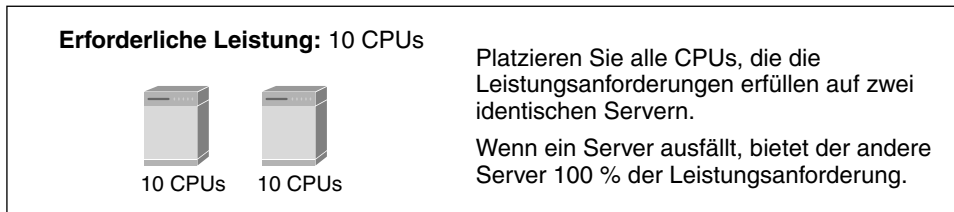
- Austausch und Neukonfiguration von Hardwarekomponenten im laufenden Betrieb
- Ausführung mehrerer Anwendungen in fehlerisolierten Domänen auf dem Server
- Aufrüstung hinsichtlich Kapazität und Leistungsgeschwindigkeit sowie E/A-Konfiguration ohne Neustart des Systems

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 Einzelpunkt-Versagen 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

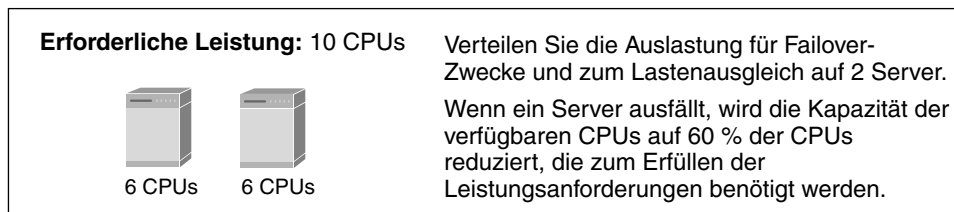


Die Rechnerkapazität der in [Abbildung 5-3](#) oben dargestellten Server ist identisch. Die Leistungsanforderungen werden von einem Server allein gehandhabt. Der andere Server stellt 100 % Leistung bereit, wenn er als Reserve in Anspruch genommen wird.

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 nachfolgenden 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

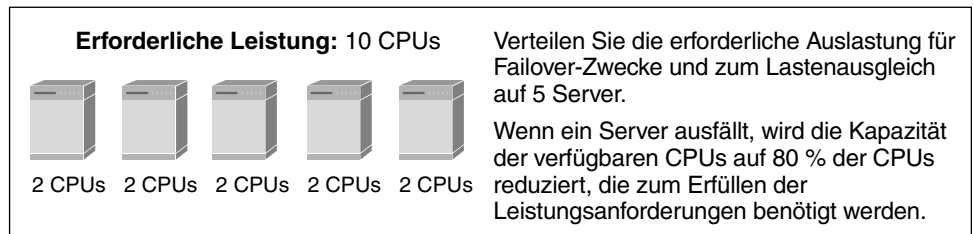


In dem System, das in [Abbildung 5-4](#) oben dargestellt ist, stehen beim Ausfall eines Servers sämtliche Dienste zur Verfügung, jedoch nur zu einem bestimmten Prozentsatz der vollen Kapazität. Der verbleibende Server stellt die Rechnerkapazität von 6 CPUs bereit, es werden also 60 % der 10-CPU-Anforderung 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 [Abbildung 5-5](#) dargestellte Konzept fünf 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 vonnöten.

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 71](#) 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 nachfolgenden Tabelle werden 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 89](#) ermittelt wurde.

Tabelle 5-7 Anpassungen von CPU-Einschätzungen für unterstützende Komponenten

Komponente	CPUs	Arbeitsspeicher
Messaging Server (MTA, Eingang)	2	4 GB
Messaging Server (MTA, Ausgang)	2	4 GB

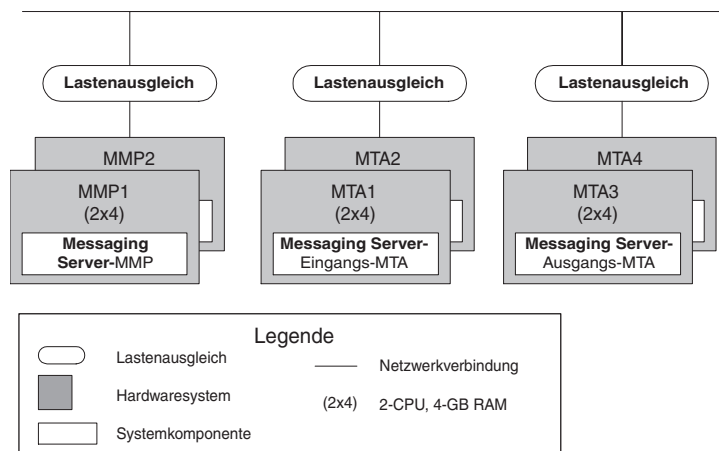
Tabelle 5-7 Anpassungen von CPU-Einschätzungen für unterstützende Komponenten

Komponente	CPUs	Arbeitsspeicher
Messaging Server (MMP)	2	4 GB
Messaging Server (Nachrichtenspeicher)	2	4 GB

Gehen Sie bei diesem Beispiel davon aus, dass in der auf die technischen Anforderungen bezogenen Phase 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 zum Ausfall der Dienstbereitstellung 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.

Zur Erfüllung der Verfügbarkeitsanforderung 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.

Abbildung 5-6 Verfügbarkeitsstrategie für Messaging Server-Beispiel

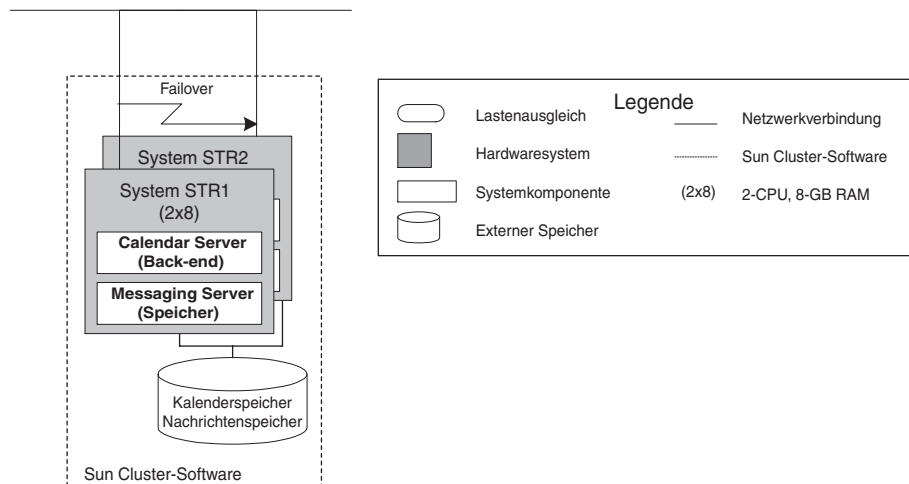
In der obigen Darstellung hat sich die Anzahl der CPUs im Vergleich zur ursprünglichen Einschätzung verdoppelt. Für die Verdopplung der CPUs gibt es folgende Gründe:

- Wenn ein Server ausfällt, stellt der verbleibende Server die CPU-Kapazität zur Handhabung der Last bereit.
- Hinsichtlich der Skalierbarkeitsanforderung, gemäß der kein einzelner Server zu mehr als 80 % der Spitzenauslastung beansprucht werden soll, wird dieser Sicherheitsspielraum durch die zusätzliche CPU-Kapazität bereitgestellt.
- Hinsichtlich der Skalierbarkeitsanforderung, gemäß der 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 nachfolgenden Abbildung ist ein Beispiel einer Failover-Strategie für das Calendar Server-Back-End und den Messaging Server-Datenspeicher dargestellt. Das Calendar Server-Back-End und der Datenspeicher 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 5-7 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:

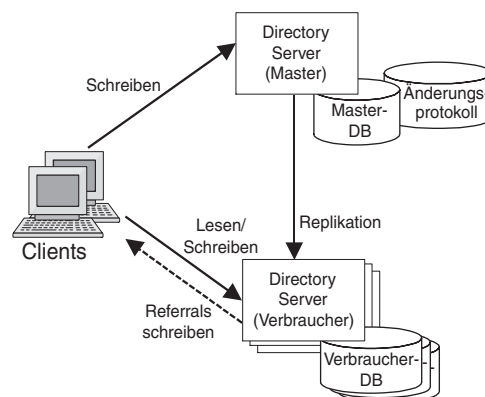
- **Mehrere Datenbanken.** Speichert unterschiedliche Bestandteile eines Verzeichnisbaums in separaten Datenbanken.
- **Vorwärtsverkettung und Referrals.** Nimmt die Verknüpfung verteilter Daten in einen einzelnen Verzeichnisbaum vor.
- **Einzelmaster-Replikation.** Stellt eine zentrale Quelle für die 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 nachfolgenden Abschnitten, „**Einzelmaster-Replikation**“ und „**Multimaster-Replikation**“, werden die grundlegenden Replikationsstrategien detailliert erläutert. Ausführliche Informationen zu den Verfügbarkeitsstrategien für Directory Server finden Sie im *Directory Server-Handbuch zur Bereitstellungsplanung*, <http://docs.sun.com/doc/817-7607>).

Einzelmaster-Replikation

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

Abbildung 5-8 Beispiel einer 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.

Hier einige Vorteile einer Einzelmaster-Replikation:

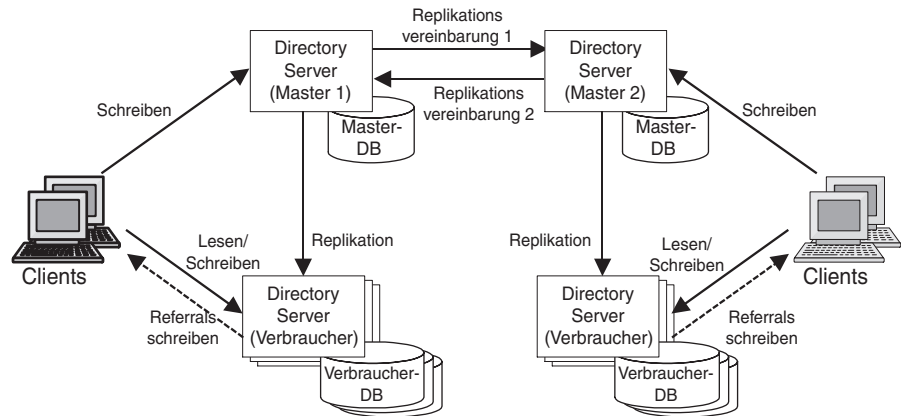
- Einzelne für datenbankbezogene Lese- und Schreibvorgänge optimierte Directory Server-Instanz
- Eine beliebige Anzahl von für Lese- und Suchvorgänge optimierten Verbraucher-Instanzen von Directory Server
- Horizontale Skalierbarkeit für Verbraucher-Instanzen von Directory Server

Multimaster-Replikation

In der nachfolgenden Abbildung ist eine Multimaster-Replikationsstrategie dargestellt, die möglicherweise zur globalen Verteilung des Verzeichniszugriffs verwendet wird.

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.

Abbildung 5-9 Beispiel einer Multimaster-Replikation



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

Ausarbeiten 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 normalerweise 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 auch überwachen, wie latente Kapazität in einem bereitgestellten System verwendet wird, 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.

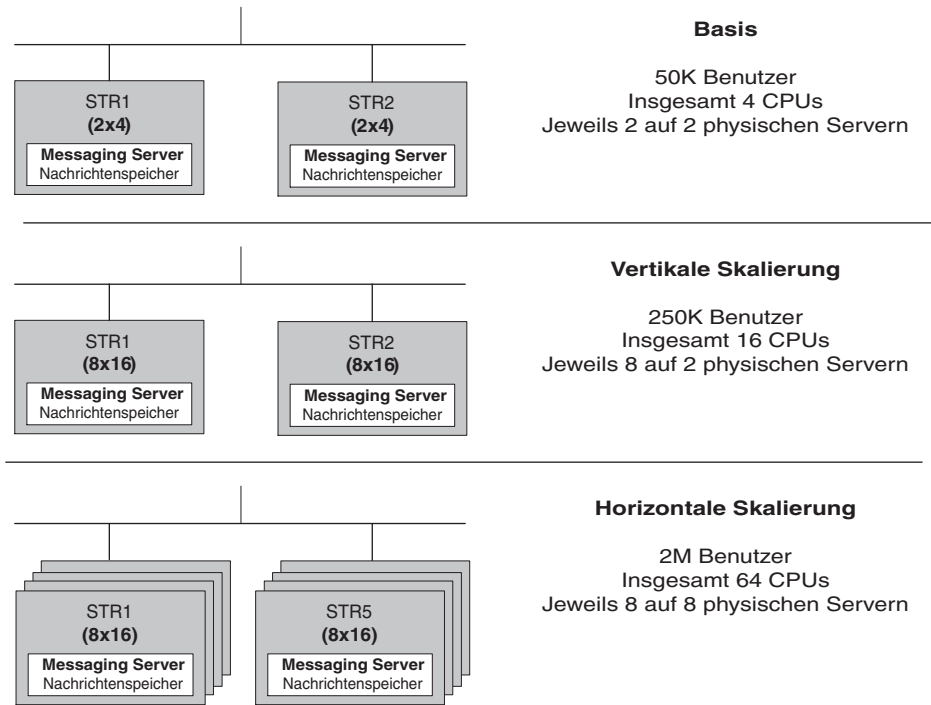
Skalierbarkeitsbeispiel

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 50000 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 250000 Benutzer und 2000000 Benutzer skaliert werden kann.

HINWEIS In [Abbildung 5-10](#) wird der Unterschied zwischen vertikaler und horizontaler Skalierung erläutert. In dieser Abbildung sind keine zusätzlichen Faktoren zu sehen, die bei der Skalierung berücksichtigt werden müssen, beispielsweise Lastenausgleich, Failover sowie Änderungen in Anwendungsmustern.

Abbildung 5-10 Beispiele für horizontale und vertikale Skalierung



Identifizieren von Leistungsengpässen

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

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

Tabelle 5-8 Datenzugriffspunkte

Hardwareklasse	Relative Zugriffsgeschwindigkeit	Lösungsvorschläge zur Leistungssteigerung
Prozessor	Nanosekunden	Vertikale Skalierung: Mehr Verarbeitungsleistung bereitstellen, Prozessorcache vergrößern Horizontale Skalierung: Parallele Verarbeitungsleistung für den Lastenausgleich bereitstellen
Systemspeicher (RAM)	Mikrosekunden	Systemspeicher bestimmten Aufgaben zuordnen Vertikale Skalierung: Zusätzlichen Speicher bereitstellen 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 Festplattenzugriff bestimmten Funktionen zuordnen, beispielsweise dem reinen Lese- oder Schreibzugriff Daten, auf die häufig zugegriffen wird, im Systempeicher zwischenspeichern
Netzwerk-schnittstelle	Variiert in Abhängigkeit von Bandbreite und Zugriffsgeschwindigkeit von Knoten im Netzwerk	Bandbreite erhöhen Beschleunigungshardware für den Transport sicherer Daten bereitstellen Leistung der Knoten im Netzwerk erhöhen, um höhere Datenverfügbarkeit zu gewährleisten

HINWEIS In [Tabelle 5-8](#) 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. An letzter Stelle steht die Prüfung der Netzwerkschnittstelle auf potenzielle Engpässe sowie die Ausarbeitung von Strategien, mit denen diese Engpässe beseitigt werden können.

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 (z. B. Festplatten) 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 Vorwärtsverkettung und Verknüpfung zur Verfügung, mit denen sich der Zugriff auf Verzeichnisdienste optimieren lässt.

Im Kapitel zum Größenaspekt des Systems im *Directory Server Deployment Planning Guide*, <http://docs.sun.com/doc/817-7607>, werden unterschiedliche Faktoren erläutert, die bei der Planung des Datenträger-(Festplatten-)Zugriffs berücksichtigt werden müssen. Folgende Themen werden in diesem Kapitel behandelt:

- **Minimum memory and disk space requirements.** Hier finden Sie Einschätzungen dazu, wie viel Festplatten- und Speicherplatz für Verzeichnisse unterschiedlicher Größe benötigt wird.
- **Sizing physical memory for cache access.** Hier finden Sie Anleitungen zur Einschätzung der Cachegröße gemäß der geplanten Nutzung von Directory Server sowie zur Planung der Gesamtspeichernutzung.
- **Sizing disk subsystems.** Hier erfahren Sie, wie die Planung der Festplattengröße gemäß den Verzeichnissuffixen vorgenommen wird und welche Directory Server-Faktoren sich auf die Datenträgernutzung 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 nachfolgenden Tabelle sind einige Systemqualitäten und die zugehörigen Überlegungen im Hinblick auf die Ressourcenverwaltung aufgeführt.

Tabelle 5-9 Überlegungen in Bezug auf die 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.)
Latente Kapazität	Kann mit Ihrer Strategie eine Last gehandhabt werden, die die Leistungseinschätzungen übersteigt? Wird die übermäßige Last durch die vertikale Skalierung auf Server, den Lastenausgleich auf andere Server oder durch beides gehandhabt? Reicht die latente Kapazität zur Handhabung ungewöhnlicher Spitzenlast aus, bis der nächste Meilenstein der Bereitstellungsskalierung erreicht wird?
Sicherheit	Haben Sie den leistungsbezogenen Mehraufwand, der für die Handhabung sicherer Transaktionen erforderlich ist, ausreichend berücksichtigt?
Verfügbarkeit	Haben Sie in Bezug auf horizontal redundante Lösungen die langfristigen Wartungskosten ausreichend berücksichtigt? Haben Sie die geplanten Ausfallzeiten berücksichtigt, die für die Wartung des Systems erforderlich sind? Wurde ein Kostengleichgewicht zwischen High-End- und Low-End-Servern erzielt?
Skalierbarkeit	Haben Sie in Ihrer Einschätzung Meilensteine für die Bereitstellungsskalierung berücksichtigt? 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?

Tabelle 5-9 Überlegungen in Bezug auf die Ressourcenverwaltung (*Fortsetzung*)

Systemqualität	Beschreibung
Wartungseignung	Haben Sie die Kosten für Verwaltung, Überwachung und Wartung in Ihr Verfügbarkeitskonzept aufgenommen? Haben Sie Lösungen für die delegierte Verwaltung (bei denen Endbenutzern die Ausführung einiger Verwaltungsaufgaben ermöglicht wird) in Erwägung gezogen, um Verwaltungskosten zu senken?

Verwalten von Risiken

Beim Großteil der Informationen, auf denen sich ein Bereitstellungskonzept stützt, beispielsweise die Dienstanforderungen 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, dass dem anfallenden Datenverkehr nicht gewachsen ist. Ein Konzept, dass 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.

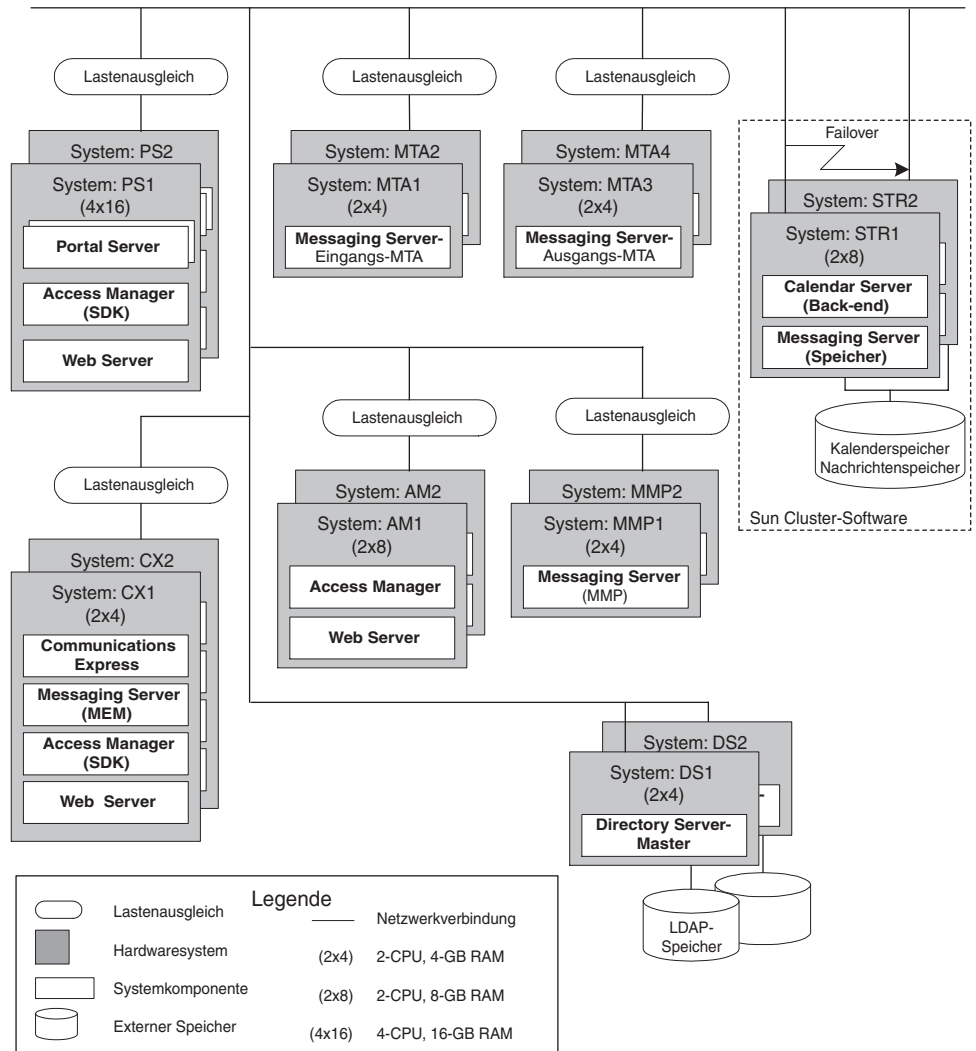
Der verschwenderische Einsatz von Ressourcen ist ein Fehler des Konzepts, da die nicht genügend beanspruchten Ressourcen an anderer Stelle hätten eingesetzt werden können. Zudem könnten verschwenderische Lösungen von den Interessengruppen als Nichterfüllung von in gutem Glauben abgeschlossenen Verträgen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

Beispiel für eine Bereitstellungsarchitektur

In der nachfolgenden 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 nachfolgenden 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 5-11 Beispiel für eine Bereitstellungsarchitektur



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. Die 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 116
- „Installation und Konfiguration der Software“ auf Seite 117
- „Entwickeln von Pilotbereitstellungen und Prototypen“ auf Seite 117
- „Entwickeln von Pilotbereitstellungen und Prototypen“ auf Seite 117
- „Testen von Pilot- und Prototypbereitstellungen“ auf Seite 118
- „Einsetzen einer Produktionsbereitstellung“ auf Seite 120

Informationen zur Implementierung des Bereitstellungskonzepts

Nachdem die Bereitstellungsarchitektur genehmigt wurde und die Implementierungsspezifikationen und -pläne abgeschlossen wurden, treten Sie in die Implementierungsphase des Lebenszyklus der Lösung ein. 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 Belastungstest 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 Benutzerhandbuchs 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 Vorgehensweisen. 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 Enterprise System-Installationshandbuch*, <http://docs.sun.com/doc/819-0805> enthält Informationen dazu, wie man Informationen für einen Installationsplan zusammenstellt. Das Installationshandbuch enthält detaillierte Konfigurationsinformationen und Arbeitsblätter, mit deren Hilfe Sie diese Informationen dokumentieren können. Außerdem enthält das Installationshandbuch Anleitungen für häufige Installationsszenarien, die mehrere Java Enterprise System-Komponenten umfassen. Weitere Informationen erhalten Sie im Abschnitt zur Installationsvorbereitung im *Java Enterprise System-Installationshandbuch*, <http://docs.sun.com/doc/819-0805>.

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. Sie können sich den ersteren Bereitstellungstyp an 80:20-Bereitstellung vorstellen (80 % der Dienste werden von Java Enterprise System zur Verfügung gestellt) und in gleicher Weise letzteren als 20:80-Bereitstellung.

Im Fall von 80:20-Bereitstellungen entwickeln Sie normalerweise in der Implementierungsphase eine Pilot-Bereitstellung 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 Echte 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 es sollten 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.

Durch 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 eine erhebliche Menge neuer 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

A

Access Manager 59, 87
80:20-Bereitstellungen 19, 118
Anwendungsanalyse 40
 Mit Auswirkung auf das
 Bereitstellungskonzept 81
Anwendungsfälle 42
 Einschätzen von Prozessoranforderungen 88
 Identitätsbasiertes Kommunikationsbeispiel 72
 Messaging Server-Beispiel 67
 Mit Auswirkung auf das
 Bereitstellungskonzept 81
Anwendungsmuster 33
Application Server 59
Architektur, dreidimensional 56

B

Beispiele
 Bereitstellungsarchitektur 112
 Directory Server 101
 Einschätzen von Prozessoranforderungen 85
 Einschätzen von Prozessoranforderungen für
 sichere Transaktionen 91
 Einzelmaster-Replikation 101
 Failover 100
 Identitätsbasierte Kommunikation 71
 Lastenausgleich 96, 98
 Logische Architektur 65
 Messaging Server, logische Architektur 66
 Multimaster-Replikation 102

 Replikation von Diensten 101
 Skalierbarkeit 105
 Verfügbarkeitskonzept 98
 Zugriffszonen 75
Belastungstests 119
Benutzerverwaltungsplan 79
Bereitstellungsarchitektur 79
 Beispiel 112
Bereitstellungskonzept
 Anwendungsanalyse 81
 Anwendungsfälle 81
 Ausgaben 79
 Dienstqualitätsanforderungen 81
 Faktoren 81
 Gesamtkosten 82
 Geschäftsziele 82
 Informationen 78
 Methodik 82
 Projektgenehmigung 78
 Prozessoranforderungen 82
 Replizieren von Diensten 83
 Vereinbarungen auf Dienstebene 81
Bereitstellungskonzeptphase 25
Bereitstellungsplanung
 Informationen 21
 Lösungslebenszyklus 22
 Schrittweise Annäherung 35
Bereitstellungsszenario 55, 76, 78
Betriebsanforderungen 32
Betriebsphase 27
Betriebsplan (Run Book) 80
Budgeteinschränkungen 37

C

- Calendar Server 59, 87
- Client-Schicht
 - Mehrschichtiges Architekturmodell 64
- Communications Express 59

D

- Datenschicht
 - Mehrschichtiges Architekturmodell 65
- Dienstebene, Anforderungen 53
- Dienstqualitätsanforderungen 43, 52
 - Funktion im Bereitstellungs-konzept 78
 - Mit Auswirkung auf das Bereitstellungs-konzept 81
- Directory Proxy Server 59, 63
- Directory Server 59, 67, 88
 - Einzelmaster-Replikation 101
 - Multimaster-Replikation 101, 102
- DMZ
 - Externe Zugriffszone 76
- Dokumentation 12
 - Installation Guide 60
 - Installationshandbuch 117
 - Technischer Überblick 19, 57, 58
 - Überblick 12
- Dreidimensionale Architektur 56

E

- Einschätzen von Prozessoranforderungen 82, 83
 - Anwendungsfälle 88
 - Beispiel 85
 - Sichere Transaktionen 90
- Einzelmaster-Replikation 101
 - Beispiel 101
- Entwurf einer mehrschichtigen Architektur 64
- Externe Zugriffszone (DMZ) 76

F

- Failover 94
 - Beispiel 100
 - Sun Cluster-Software 98
- Fehlertolerante Systeme 46
- Funktionstests 118

G

- Gesamtkosten 38
 - Mit Auswirkung auf das Bereitstellungs-konzept 82
- Geschäftliche Einschränkungen 36
 - Budgeteinschränkungen 37
 - Gesamtkosten 38
 - Migrationsprobleme 37
 - Planungsmandate 37
- Geschäftsanforderungen
 - Anwendungsmuster 33
 - Betriebsanforderungen 32
 - definieren 30
 - Geschäftsziele 30
 - Kenntnisse über Benutzer 31
 - Regulierungsanforderungen 34
 - Sicherheitsziele 34
 - Unternehmenskultur 33
 - Vereinbarungen auf Dienstebene 36
- Geschäftsdienstschicht
 - Mehrschichtiges Architekturmodell 65
- Geschäftsziele
 - definieren 30
 - Mit Auswirkung auf das Bereitstellungs-konzept 82

H

- Horizontal redundante Systeme 96

I

- Identifizieren von Engpässen
 - Bereitstellungskonzept 83
- Identitätsbasiertes Kommunikationsbeispiel 71
 - Anwendungsfälle 72
 - Einschätzen von Prozessoranforderungen 85
- Implementierungsphase 26, 118
 - Entwickeln von Pilotbereitstellungen und Prototypen 117
 - Informationen 116
- Implementierungspläne 79
- Implementierungsspezifikationen 79
- Installation von Java Enterprise System 117
- Installationsplan 79
- Instant Messaging 59
- Interne Zugriffszone (Intranet) 76

J

- Java Enterprise System
 - 20:80-Bereitstellungen 19
 - 80:20-Bereitstellungen 19
 - Benutzerdefinierte Dienste 19
 - Dienste 19
 - Dreidimensionale Architektur 56
 - Einsetzen einer Produktionsbereitstellung 120
 - Informationen 17
 - Installation 117
 - Komponenten 58
 - Komponentenabhängigkeiten 59
 - Migrationsprobleme 21
 - Systemdienste 18
 - Zugriffskomponenten 63

K

- Komponentenabhängigkeiten 59
 - Webcontainer-Unterstützung 62

L

- Lastenausgleich 94
 - Beispiel 96
- Latente Kapazität 52
 - Skalierbarkeitsüberlegungen 104
- Leistung
 - Dienstqualitätsanforderung 45
 - Identifizieren von Engpässen 106
 - Optimieren von Ressourcen 110
- Logische Architektur
 - Beispiele 65
 - Entwurf 57
 - Identitätsbasiertes Kommunikationsbeispiel 71
- logische Architektur
 - Mit Auswirkung auf das Bereitstellungskonzept 81
- Logische Architekturen 55
- Logische Schichten
 - Mehrschichtiges Architekturmodell 64
- Logisches Konzept
 - Informationen 55
- Lösungslebenszyklus 22
 - Bereitstellungskonzeptphase 25, 78
 - Betriebsphase 27
 - Implementierungsphase 26, 116
 - Phase der Geschäftsanalyse 23, 29
 - Phase des logischen Konzepts 24, 55
 - Technische Anforderungen, Phase 23, 39

M

- Message Queue 59
- Messaging Server 59
 - Anwendungsfälle 67
 - Beispiel für logische Architektur 66
 - Lastenausgleichsbeispiel 98
 - Logisch eindeutige Dienste 62
 - Message Multiplexor (MMP) 63, 67, 87
 - Message Transfer Agent (MTA) 62, 67
 - Messenger Express Multiplexor (MEM) 63
 - Nachrichtenspeicher (STR) 62, 67, 87

- Migrationsplan 79
- Migrationsprobleme 21
 - Als geschäftliche Einschränkung 37
- Multimaster-Replikation 101
 - Beispiel 102

N

- n+1-Failover-System 96

O

- Optimieren von Ressourcen
 - Bereitstellungskonzept 83
- Optimierung
 - Festplattenzugriff 108
 - Ressourcennutzung 109

P

- Phase der Geschäftsanalyse 23
 - Informationen 29
- Phase des logischen Konzepts 24
- Pilotbereitstellungen 117
- Pilottests
 - Testen 118
- Planungsmandate 37
- Portal Server 60, 63
 - Mobile Access 63
 - Secure Remote Access 60, 63
- Präsentationsschicht
 - Mehrschichtiges Architekturmodell 64
- Projektgenehmigung 78
- Prototypen 117
 - Testen 118
- Prozessoranforderungen
 - Einschätzung 83

Q

- QoS (Dienstqualitätsanforderungen) 43

R

- Regulierungsanforderungen 34
- Replikation von Diensten
 - Directory Server-Beispiel 101
 - Verfügbarkeitsstrategie 95
- Replizieren von Diensten 83
- Risikoverwaltung 111
- Roll-out-Plan 80
- Run Book 80

S

- Schulungsplan 80
- Sichere Zugriffszone 76
- Sicherheit
 - Dienstqualitätsanforderung 50
 - Einschätzen von Prozessoranforderungen 83
 - Optimieren von Ressourcen 110
- Skalierbarkeit
 - Beispiel 105
 - Dienstqualitätsanforderung 48
 - Optimieren von Ressourcen 110
 - Strategien 103
 - Wachstumseinschätzung 49
- SLA 36
- Solaris
 - Patches 14
 - Support 14
- Sun Cluster-Software 98
 - Failover-Beispiel 100
- Support
 - Solaris 14

T

- Technische Anforderungen
 - Dienstebene, Anforderungen 53
 - Latente Kapazität 52
 - Leistung 45
 - Sicherheit 50
 - Skalierbarkeit 48
 - Verfügbarkeit 46
 - Zweckmäßigkeit 52
- Technische Anforderungen, Phase 23
 - Anwendungsanalyse 40
 - Anwendungsfälle 42
 - Dienstqualitätsanforderungen 43
 - Informationen 39
- Testen
 - Pilottests und Prototypen 118
- Testplan 80
- Tests
 - Belastungstests 119
 - Funktionstests 118

U

- Unternehmenskultur 33

V

- Vereinbarungen auf Dienstebene 36
 - Anforderungen 53
 - Mit Auswirkung auf das
Bereitstellungskonzept 81
- Verfügbarkeit
 - Beispiele 98
 - Dienstqualitätsanforderung 46
 - Failover 94
 - Horizontal redundante Systeme 96
 - Lastenausgleich 94
 - n+1-Failover-System 96
 - Optimieren von Ressourcen 110
 - Priorisieren 47
 - Replikation von Diensten 95

- Verfügbarkeitsstrategien
 - Ermittlung 94
- Verwalten von Risiken 111
 - Bereitstellungskonzept 83

W

- Wartungseignung
 - Optimieren von Ressourcen 111
- Web Server 60, 88
- Wiederherstellungsplan 80

Z

- Zugriffszonen 75
- 20:80-Bereitstellungen 19
 - Implementierungsphase 118
- Zweckmäßigkeit
 - Dienstqualitätsanforderung 52

