Sun Java Enterprise System 5 – Technische Übersicht

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

Teilenr.: 820–0887
März 2007

070426@17039

Inhalt

Tabellen

Abbildungen

Vorwort

Das Handbuch *Sun Java Enterprise System 5 – Technische Übersicht* enthält eine Einführung in die konzeptionelle Basis von Sun Java™ Enterprise System (Java ES). Darüber hinaus werden darin die Komponenten, die Architektur, die Vorgänge und die Funktionen von Java ES beschrieben.

Dieser Überblick beschreibt die technischen Konzepte und die Terminologie, die im Java ES-Dokumentationssatz verwendet werden. Die wichtigsten technischen Begriffe werden im letzten Abschnitt eines jeden Kapitels erläutert.

An wen richtet sich dieses Handbuch

Dieses Handbuch ist für Benutzer gedacht, die Softwarelösungen auf Basis von Java ES entwerfen, bereitstellen oder verwalten, wie beispielsweise Geschäftsanalysten, Systemarchitekturdesigner, Außendiensttechniker und Systemadministratoren.

Die Leser dieses Handbuchs sollten mit folgenden Technologien vertraut sein:

- Allgemeine Netzwerkkonzepte
- Sicherheitsgrundlagen für Authentifizierung und Autorisierung
- Programmiersprache Java
- Komponenten von Java 2 Platform, Standard Edition (J2SE™-Plattform) und Komponenten von Java 2 Platform, Enterprise Edition (J2EE™-Plattform)

Aufbau dieses Handbuchs

Dieses Handbuch ist in folgende Kapitel unterteilt:

- Kapitel 1 stellt Java ES und die Aufgaben vor, die bei Verwendung des Systems ausgeführt werden.
- Kapitel 2 beschreibt ein Architektur-Framework für die Entwicklung von Java ES-Lösungsarchitekturen und liefert eine Beispielarchitektur auf Basis dieses Frameworks.
- Kapitel 3 liefert Informationen über Funktionen, die bei der Integration von Java ES-Komponenten in ein einzelnes Softwaresystem eine entscheidende Rolle spielen.

- Kapitel 4 beschreibt die Konzepte und die Terminologie, die für jede Phase des Java ES-Lösungslebenszyklus relevant sind.
- Anhang A enthält eine Liste von Java ES-Komponenten.

Java Enterprise System-Dokumentationssatz

Der Java ES-Systemdokumentationssatz beschreibt die Bereitstellungsplanung und die Installation des Systems. Der URL für die Systemdokumentation lautet `http://docs.sun.com/coll/1286.2`. Die in der folgenden Tabelle aufgelisteten Handbücher bieten eine Einführung in Java ES. Lesen Sie die Handbücher in der hier gezeigten Reihenfolge. Java ES-Informationen und -Ressourcen sind auch unter `http://www.sun.com/bigadmin/hubs/javaes/` verfügbar.

TABELLE P–1 Java Enterprise System-Dokumentation

Titel des Dokuments	Inhalt
Sun Java Enterprise System 5 Versionshinweise für UNIX *Sun Java Enterprise System 5 Release Notes for Microsoft Windows*	Enthält die neuesten Informationen über Java ES, einschließlich bekannter Probleme. Zudem verfügen die Komponenten über ihre eigenen Versionshinweise, die in der Sammlung der Versionshinweise aufgeführt sind (`http://docs.sun.com/coll/1315.2`).
Sun Java Enterprise System 5 – Technische Übersicht	Bietet eine Einführung in die technischen Grundlagen und in das Konzept von Java ES. Beschreibt die Komponenten, die Architektur sowie Prozesse und Funktionen.
Sun Java Enterprise System Deployment Planning Guide	Bietet eine Einführung in die Planung und die Konzeptentwicklung von Bereitstellungslösungen für Unternehmen, die auf Java ES basieren. Beschreibt die Grundlagen und Prinzipien der Bereitstellungsplanung und des Bereitstellungskonzepts sowie den Lebenszyklus einer Lösung und bietet erstklassige Beispiele und Strategien, die Sie bei der Planung der unternehmensweiten Bereitstellung von Lösungen mit Java ES anwenden können.
Sun Java Enterprise System 5 – Handbuch zur Installationsplanung	Unterstützt Sie bei der Entwicklung der Implementierungsspezifikationen für die Hardware, das Betriebssystem sowie für die Netzwerkaspekte Ihrer Java ES-Bereitstellung. Beschreibt Probleme, wie beispielsweise Komponentenabhängigkeiten, die Sie in Ihrem Installations- und Konfigurationsplan berücksichtigen müssen.
Sun Java Enterprise System 5 Installationshandbuch für UNIX *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*	Führt Sie durch den Installationsprozess von Java ES. Bietet außerdem Anweisungen zur Konfiguration der Komponenten nach der Installation und zur Überprüfung der richtigen Funktionsweise der Komponenten.

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

Titel des Dokuments	Inhalt
Sun Java Enterprise System 5 Installationsanweisungen für UNIX	Liefert zusätzliche Informationen zu Konfigurationsparametern, liefert die für die Planung Ihrer Konfiguration benötigten Arbeitsblätter und führt Referenzmaterial auf, wie beispielsweise Standardverzeichnisse und Anschlussnummern auf dem Solaris™-Betriebssystem (Solaris OS) und der Linux-Betriebsumgebung.
Sun Java Enterprise System 5 - Aufrustungshandbuch fur UNIX	Liefert Anweisungen zum Aufrüsten von zuvor installierten Versionen auf Java ES 5.
Sun Java Enterprise System 5 Upgrade Guide for Microsoft Windows	
Sun Java Enterprise System 5 Überwachungshandbuch	Liefert Anweisungen zum Einrichten des Monitoring Frameworks für jede Produktkomponente und zur Verwendung der Monitoring Console für die Anzeige von Echtzeit-Daten und das Erstellen von Überwachungsregeln.
Sun Java Enterprise System Glossary	Definiert die in der Java ES-Dokumentation verwendeten Begriffe.

Typografische Konventionen

In der nachfolgenden Tabelle werden die typografischen Konventionen beschrieben, die in diesem Handbuch verwendet werden.

TABELLE P–2 Typografische Konventionen

Schriftart	Bedeutung	Beispiel
`AaBbCc123`	Befehlsnamen, Dateinamen, Verzeichnisnamen und Bildschirmcomputerausgaben	Bearbeiten Sie Ihre `.login` -Datei. Verwenden Sie `ls -a`, um sämtliche Dateien aufzuführen. `machine_name% you have mail.`
`AaBbCc123`	Eingabe durch den Benutzer (im Gegensatz zur Computerausgabe auf dem Bildschirm	`machine_name%` **`su`** `Password:`
AaBbCc123	Ein Platzhalter, der durch einen echten Namen oder Wert ersetzt werden soll.	Der Befehl zum Entfernen von Dateien lautet `rm` *filename* .

TABELLE P–2 Typografische Konventionen *(Fortsetzung)*

Schriftart	Bedeutung	Beispiel
AaBbCc123	Buchtitel, neue Begriffe und hervorzuhebende Begriffe (Beachten Sie, dass einige hervorgehobene Begriffe online fett angezeigt werden)	Lesen Sie Kapitel 6 im *Benutzerhandbuch*. Ein *cache* ist eine Kopie, die lokal gespeichert wird. Speichern Sie die Datei *nicht*.

Dokumentation, Support und Schulung

Die Sun-Website liefert Informationen zu den folgenden zusätzlichen Ressourcen:

- Dokumentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Schulung (`http://www.sun.com/training/`)

Suchen nach Sun-Produktdokumentationen

Neben der Suche nach Sun-Produktdokumentationen über die Website docs.sun.com[SM] können Sie eine Suchmaschine verwenden, bei der Sie folgende Syntax in das Suchfeld eingeben:

search-term `site:docs.sun.com`

Um beispielsweise nach dem Begriff "Broker„ zu suchen, geben Sie Folgendes ein:

```
broker site:docs.sun.com
```

Um weitere Sun-Websites in Ihre Suche einzubeziehen (z. B. java.sun.com, www.sun.com und developers.sun.com), geben Sie `sun.com` statt `docs.sun.com` in das Suchfeld ein.

Verweise auf Drittanbieter-Websites

In der vorliegenden Dokumentation wird auf URLs von Drittanbietern verwiesen, über die zusätzliche relevante Informationen zur Verfügung gestellt werden.

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

Kommentare sind willkommen

Sun möchte seine Dokumentation laufend verbessern. Ihre Kommentare und Vorschläge sind daher immer willkommen. Wenn Sie Kommentare abgeben möchten, rufen Sie die Seite `http://docs.sun.com` und klicken Sie auf "Kommentare senden". Geben Sie im Online-Formular den Dokumenttitel und die Teilenummer an. Die Teilenummer ist eine sieben- oder neunstellige Zahl, die Sie auf der Titelseite des Buchs oder im Dokument-URL finden. Die Teilenummer dieses Buches lautet z. B. 820–0167–10.

◆ ◆ ◆ KAPITEL 1

1 Einführung in Java Enterprise System

Sun Java™ Enterprise System (Java ES) besteht aus einem Satz von Softwarekomponenten, die Dienste bieten, die bei der Unterstützung verteilter Anwendungen in Unternehmensstärke in einer Netzwerk- oder Internetumgebung benötigt werden. Solche Anwendungen werden als verteilte Unternehmensanwendungen bezeichnet. In diesem Handbuch werden die Softwarekomponenten von Java ES und die von ihnen bereitgestellten Dienste beschrieben.

In diesem Kapitel werden das System Java ES und die Aufgaben vorgestellt, die bei Verwendung des Systems ausgeführt werden. Das Kapitel enthält die folgenden Abschnitte:

Warum Sie Java ES benötigen

Die Geschäftsanforderungen von heute fordern Softwarelösungen, die über eine Netzwerk- oder Internetumgebung hinweg verteilt sind sowie hohe Leistung, Verfügbarkeit, Sicherheit, Skalierbarkeit und Zweckmäßigkeit bieten.

Java ES bietet Infrastrukturdienste, die benötigt werden, um verteilte Unternehmensanwendungen zu unterstützen, die in der Regel folgende Merkmale aufweisen:

- **Verteilt:** Die Anwendung besteht aus interagierenden Software- komponenten, die über eine vernetzte Umgebung verteilt werden, die geografisch entfernte Standorte umfassen kann. Diese verteilten Komponenten, die auf den verschiedenen Computern der Umgebung ausgeführt werden, arbeiten zusammen, um spezielle Geschäftsfunktionen für die end users und andere Geschäftsanwendungen bereitzustellen.

- **Unternehmensstärke:** Umfang und Größe der Anwendung erfüllen die Anforderungen einer Produktionsumgebung oder eines Internetdienstanbieters. Die Anwendung erstreckt sich in der Regel über ein gesamtes Unternehmen und integriert zahlreiche Abteilungen, Einsatzbereiche und Vorgänge in einem einzigen Softwaresystem. Die Anwendung muss hohe Dienstqualitätsanforderungen in Bezug auf die Leistung, Verfügbarkeit, Sicherheit, Zweckmäßigkeit und Skalierbarkeit erfüllen.

Verteilte Unternehmensanwendungen erfordern eine zugrunde liegende Infrastruktur von Diensten, die den verteilten Komponenten die Kommunikation untereinander, die Koordination der Arbeit, die Implementierung eines sicheren Zugriffs usw. ermöglichen. Diese Infrastrukturdienste werden von einer Hardwareumgebung aus Computern und Netzwerkverbindungen unterstützt. Diese Umgebung enthält die Hardware-Architekturen SPARC® und x86 (Intel und AMD).

In der folgenden Abbildung ist das allgemeine Ebenenschema dargestellt. Java ES stellt überwiegend die Ebene der verteilten Infrastrukturdienste bereit, die in der Abbildung dargestellt sind.

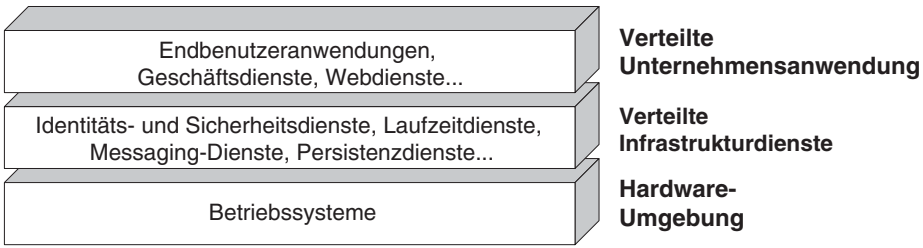

ABBILDUNG 1–1 Erforderliche Unterstützung für verteilte Unternehmensanwendungen

Zu den von Java ES bereitgestellten Diensten gehören:

- **Zugangsdienste:** Über diese Dienste können Mitarbeiter, Telearbeiter, Knowledge Worker, Geschäftspartner, Lieferanten und Kunden von innerhalb und außerhalb des Unternehmensnetzwerks auf die Unternehmensressourcen zugreifen. Diese Dienste bieten Benutzer-Communities jederzeit und von beliebigen Standorten Zugriff und sorgen für personalisierte Integration, Aggregation, Sicherheit, mobilen Zugriff und Suchfunktionen.
- **Kommunikations- und Zusammenarbeitsdienste:** Diese Dienste ermöglichen den sicheren Austausch von Informationen innerhalb verschiedener Benutzer-Communities. Im Kontext der Unternehmensumgebung des jeweiligen Benutzers stehen spezifische Funktionen, wie beispielsweise Messaging, Zusammenarbeit in Echtzeit sowie Instant Messaging und Konferenz- oder Kalenderplanungsfunktionen zur Verfügung.

Hinweis – Dieses Handbuch bezieht sich auf Komponenten der Sun Java Communications Suite, die von Java ES-Komponenten abhängig sind und innerhalb der Java ES-Bereitstellungsarchitektur verwendet werden. Kommunikations- und Zusammenarbeitskomponenten sind nicht in Java ES enthalten.

- **Netzwerkidentitäts- und Sicherheitsdienste:** Diese Dienste verbessern die Sicherheit und den Schutz wichtiger Unternehmensinformationen, indem sie auf globaler Basis die Durchsetzung entsprechender Zugriffssteuerungsrichtlinien über alle Communities, Anwendungen und Dienste hinweg gewährleisten. Diese Dienste verwenden ein zentrales Repository zum Speichern und Verwalten von Identitätsprofilen, Zugriffsrechten sowie von Informationen zu Anwendungen und Netzwerkressourcen.
- **Webcontainer- und Anwendungsdienste:** Diese Dienste sorgen dafür, dass verteilte Komponenten während der Laufzeit miteinander kommunizieren können und unterstützen die Entwicklung, Bereitstellung und Verwaltung von Anwendungen für eine Vielzahl von Servern, Clients und Geräten. Diese Dienste basieren auf J2EE-Technologie.

Java ES bietet außerdem Dienste, mit denen die Verfügbarkeit, Skalierbarkeit, Zweckmäßigkeit und weitere Anwendungs- oder Systemqualitäten verbessert werden. Zu den von Java ES bereitgestellten Dienstqualitätsfunktionen gehören:

- **Verfügbarkeitsdienste:** Diese Dienste sorgen für eine nahezu kontinuierliche Verfügbarkeit und bieten die Anwendungs- und Infrastrukturkomponenten, die diese unterstützen.
- **Zugriffsdienste:** Diese Dienste bieten einen internet- oder browserbasierten Zugriff auf Java ES-Dienste.
- **Überwachungsdienste.** Diese Dienste liefern Echtzeit-Informationen zu Java ES-Komponenten.

Sie können einen oder mehrere Java ES-Dienste bereitstellen, von denen jeder mehrere Java ES-Komponenten enthalten kann.

Java ES-Komponenten

Java ES stellt eine Integration verschiedener unabhängiger Softwareprodukte und -komponenten in einem einzigen Softwaresystem dar. Diese Integration wird durch eine Reihe von Funktionen auf Systemebene wie im Folgenden beschrieben ermöglicht:

- Alle Komponenten werden mit einem Satz gemeinsam genutzter Bibliotheken synchronisiert.
- Alle Java ES-Komponenten werden mithilfe eines einzigen Installationsprogramms installiert.

- Alle Java ES-Komponenten können gemeinsam ein integriertes Benutzeridentitäts- und Sicherheitsverwaltungssystem nutzen.
- Sämtliche Java ES-Komponenten verfügen über ein gemeinsames Überwachungs-Framework.

Diese Funktionen werden in den nachfolgenden Kapiteln dieses Buchs beschrieben. In diesem Abschnitt werden hauptsächlich die in Java ES integrierten Komponenten vorgestellt. Diese Systemkomponenten können in drei Hauptkategorien unterteilt werden, wie in der folgenden Abbildung gezeigt:

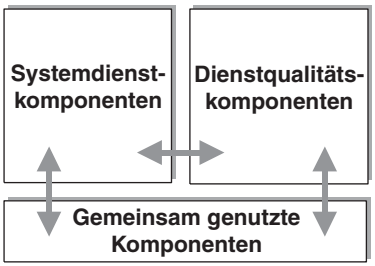

ABBILDUNG 1–2 Kategorien der Java ES-Komponenten

Die einzelnen Komponenten bieten folgende Dienste:

- **Systemdienstkomponenten:** Diese Komponenten bieten die grundlegenden Java ES-Infrastrukturdienste, die für die Unterstützung verteilter Unternehmensanwendungen benötigt werden.
- **Dienstqualitäts- komponenten:** Diese Komponenten verbessern die Verfügbarkeit, Sicherheit, Skalierbarkeit, Zweckmäßigkeit sowie andere Bereiche der Systemdienstkomponenten und verteilten Anwendungskomponenten.
- **Gemeinsam genutzte Komponenten:** Diese Komponenten bilden die Umgebung, in der viele Systemdienst- und Dienstqualitätskomponenten ausgeführt werden.

Eine Liste der Java ES-Komponenten finden Sie unter Anhang A.

Systemdienst- komponenten

Einige Java ES-Komponenten bilden die Hauptdienste für die Unterstützung verteilter Softwarelösungen. Diese Systemdienste beinhalten Zugangsdienste, Identitäts- und Sicherheitsdienste, Webcontainerdienste, J2EE-Anwendungsdienste und Persistenzdienste.

Die Systemdienstkomponenten, die diese verteilten Dienste bereitstellen und die von ihnen bereitgestellten Dienste werden in folgender Tabelle alphabetisch aufgeführt und kurz beschrieben. Jede Systemdienstkomponente ist ein Servervorgang mit mehreren Threads, der eine Vielzahl von Clients unterstützen kann. Weitere Informationen zu den einzelnen Komponenten finden Sie unter„Systemdienstkomponenten” auf Seite 71.

TABELLE 1–1 Java ES-Systemdienstkomponenten

Komponente	Bereitgestellte Systemdienste
Sun Java System Access Manager	Stellt Dienste für die Zugriffsverwaltung und die digitale Identitätsverwaltung bereit. Zu den Zugriffsverwaltungsdiensten zählen die Authentifizierung (einschließlich Single-Sign-On) und die rollenbasierte Autorisierung für den Zugriff auf Anwendungen und/oder Dienste. Die Verwaltungsdienste umfassen die zentralisierte Verwaltung einzelner Benutzerkonten, Rollen, Gruppen und Richtlinien.
Sun Java System Anwendungsserver	Stellt J2EE-Containerdienste für Enterprise JEnterprise JavaBeans™-Komponenten (EJB-Komponenten) wie Sitzungs-Beans, Einheiten-Beans und nachrichtengesteuerte Beans bereit. Der Container bietet die Infrastrukturdienste, die für die Interaktion von eng miteinander verknüpften verteilten Komponenten erforderlich sind, und macht Anwendungsserver so zu einer Plattform für die Entwicklung und Ausführung von E-Commerce-Anwendungen und Webdiensten. Anwendungsserver stellt außerdem Webcontainterdienste bereit.
Sun Java System Directory Server	Stellt ein zentrales Repository zum Speichern und Verwalten von Intranet- und Internetinformationen, wie Identitätsprofile (Mitarbeiter, Kunden, Lieferanten usw.), Benutzeranmeldeinformationen (öffentliche Schlüsselzertifikate, Passwörter und PIN-Nummern), Zugriffsrechte sowie Anwendungs- und Netzwerkressourceninformationen bereit.
Java DB[1]	Bietet eine leichtgewichtige Datenbank für die Entwicklung von Java-Anwendungen. Java DB ist die unterstützte Verteilung der Open Source-Datenbank Apache Derby mit 100 % Java-Technologie.
Sun Java System Message Queue	Bietet zuverlässigen, asynchronen Messaging-Austausch zwischen lose miteinander verknüpften verteilten Komponenten und Anwendungen. Message Queue implementiert die Java Message Service (JMS) API-Spezifikation und liefert Unternehmensfunktionen wie Sicherheit, Skalierbarkeit und Remoteverwaltung.
Sun Java System Portal Server	Stellt browserbasierten Clients, die auf Geschäftsanwendungen oder -dienste zugreifen, wichtige Zugangsdienste wie die Aggregation und Personalisierung von Inhalten bereit. Portal Server stellt darüber hinaus eine konfigurierbare Suchmaschine bereit.
Sun Java System Service Registry	Bietet eine Registry und ein Repository für die Unterstützung von Service-Oriented Architecture (SOA)-Anwendungen. Service Registry implementiert Industriestandards für die Registrierung und das Auffinden von Webdiensten sowie für die Verwaltung der dazugehörigen Informationen und Fakten, Artefakte (z. B. XML-Schemata, Unternehmensprozessregeln, Zugriffssteuerung, Versionskontrolle).

[1] Java ES 5 ist die erste Version, in der Java DB als Produktkomponente enthalten ist. Java DB wurde zunächst als gemeinsam genutzte Komponente mit dem Namen Derby Database veröffentlicht und in Java ES 2005Q4 integriert.

TABELLE 1–1 Java ES-Systemdienstkomponenten *(Fortsetzung)*

Komponente	Bereitgestellte Systemdienste
Sun Java System Web Server	Bietet J2EE -Webcontainer-Dienste, wie beispielsweise Java ™ Servlet- und JavaServer Pages™ (JSP™)-Komponenten. Web Server unterstützt zudem andere Webanwendungstechnologien für die Bereitstellung von statischem und dynamischem Webinhalt wie CGI-Skripts und Sun Java™ System Active Server Pages.

Dienstqualitätskomponenten

Neben den in Tabelle 1–1 gezeigten Systemdienstkomponenten enthält Java ES eine Vielzahl von Komponenten, mit denen die Qualität der von den Systemdienstkomponenten bereitgestellten Dienste verbessert wird. Dienstqualitätskomponenten können auch kundenspezifisch entwickelte Anwendungsdienste verbessern. Die Dienstqualitätskomponenten lassen sich in folgende Kategorien unterteilen:

- Verfügbarkeitskomponenten
- Zugriffskomponenten
- Überwachungskomponenten

Verfügbarkeitskomponenten

Verfügbarkeitskomponenten sorgen für eine nahezu kontinuierliche Systembetriebszeit von Systemdienstkomponenten und benutzerdefinierten Anwendungsdiensten. Die folgende Tabelle enthält die in Java ES enthaltenen Verfügbarkeitskomponenten und die von diesen angebotenen Dienste. Weitere Informationen zu den einzelnen Komponenten finden Sie unter „Verfügbarkeitskomponenten” auf Seite 76.

TABELLE 1–2 Java ES-Verfügbarkeitskomponenten

Komponente	Bereitgestellte Verfügbarkeitsdienste
High Availability Session Store	Bietet einen Datenspeicher, der Anwendungsdaten, insbesondere Sitzungsstatusdaten, auch im Fehlerfall verfügbar macht.
Sun Cluster	Bietet Hochverfügbarkeits- und Skalierbarkeitsdienste für Java ES und für die Anwendungen, die basierend auf der Java ES-Infrastruktur ausgeführt werden, sowie für die Hardwareumgebung, in der beide bereitgestellt werden.
Sun Cluster Geographic Edition[1]	Schützt Anwendungen mithilfe von mehreren geografisch getrennten Clustern und einer redundanten Infrastruktur, durch die Daten zwischen diesen Clustern repliziert werden, vor unerwarteten Unterbrechungen. Die Software Sun Cluster Geographic Edition ist eine geschichtete Erweiterung der Sun Cluster-Software.

[1] Java ES 5 ist die erste Version, die Sun Cluster Geographic Edition als Java ES-Produktkomponente enthält.

Zugriffskomponenten

Zugriffskomponenten bieten Front-End-Zugriff auf Systemdienste und sorgen für einen sicheren Zugriff über Internetstandorte, die sich außerhalb der Firewall des Unternehmens befinden. Neben diesen Zugriffsmöglichkeiten bieten viele auch eine Routing- und Speicherfunktion. Die folgende Tabelle enthält die in Java ES enthaltenen Zugriffskomponenten und die von diesen angebotenen Dienste. Weitere Informationen zu den einzelnen Komponenten finden Sie unter „Zugriffskomponenten" auf Seite 78.

TABELLE 1–3 Java ES-Zugriffskomponenten

Komponente	Bereitgestellte Zugriffsdienste
Sun Java System Portal Server (beinhaltet Secure Remote Access)	Bietet sicheren Internetzugriff von außerhalb der Unternehmens-Firewall auf den Inhalt und die Dienste von Portal Server, einschließlich interner Portale.
Sun Java System Web Proxy Server	Übernimmt Cache- und Filterfunktionen sowie die Verteilung von Webinhalt sowohl für ausgehende als auch für eingehende Internetanforderungen.

Überwachungskomponenten

Java ES enthält eine neue Überwachungsfunktion, die Echtzeit-Systemstatus und anpassbare Überwachungsaufträge bereitstellt. Die Überwachung wird von Sun Java System Monitoring Console Produktkomponente implementiert, die vom Sun Java System Monitoring Framework Gemeinsam genutzte Komponente unterstützt wird. Weitere Informationen finden Sie im Abschnitt „Überwachungskomponenten" auf Seite 79.

Gemeinsam genutzte Komponenten

Java ES enthält eine Vielzahl an lokal installierten gemeinsam genutzten Bibliotheken, von denen viele Systemdienstkomponenten und Dienstqualitätskomponenten abhängen. Die gemeinsam genutzten Komponenten von Java ES bieten lokale Dienste für Java ES-Produktkomponenten, die auf demselben Hostcomputer ausgeführt werden.

Gemeinsam genutzte Komponenten werden oft eingesetzt, um die Portierbarkeit zwischen unterschiedlichen Betriebssystemen zu gewährleisten. Beispiele für gemeinsam genutzte Komponenten von Java ES sind unter anderem: Java 2 Platform, Standard Edition (J2SE), Netscape Portable Runtime (NSPR), Network Security Services (NSS), Java Security Services for Java (JSS) usw. Eine vollständige Liste finden Sie unter „Freigegebene Komponenten" auf Seite 80.

Die gemeinsam genutzten Komponenten werden automatisch vom Java ES-Installationsprogramm entsprechend der zu installierenden Systemdienste und Dienstqualitätskomponenten installiert.

In Sun Java Suites enthaltene Komponenten

Java ES ist sowohl als einzelne End-to-End-Infrastruktur-Softwareverteilung als auch in Form von einzelnen Suite-Verteilungen verfügbar, die auf kritische Geschäftsbedürfnisse ausgelegt sind. Java ES beinhaltet sämtliche Java ES-Komponenten, während Sun Java System-Suites Teile dieser Komponenten enthält, die ausgewählt wurden, um bestimmte Bedürfnisse im Geschäftsleben zu erfüllen. Die Java ES-Installations- und Deinstallationsprogramme sind zwar in sämtlichen Suite-Verteilungen enthalten, sie sind jedoch insofern abgespeckt, dass Sie nur die Komponenten innerhalb der Suite bearbeiten. Sämtliche gemeinsam genutzten Komponenten sind auch in sämtlichen Suite-Verteilungen enthalten.

Der Inhalt der einzelnen Suite und die Geschäftsanforderungen, die jede Suite erfüllen soll, werden in folgender Tabelle aufgeführt.

TABELLE 1–4 In Sun Java Suites enthaltene Komponenten

Suite	Geschäftsanforderung	Inhalt
Sun Java Application Platform Suite	Entwicklung, Bereitstellung und Verwaltung von dienstorientierten Architekturen (Service-oriented Architectures, SOA) der nächsten Generation	Access Manager Anwendungsserver Directory Server HADB Java DB Message Queue Monitoring Console Portal Server (enthält Secure Remote Access und Mobile Access) Service Registry Web Proxy Server Web Server
Sun Java Availability Suite	Notfallwiederherstellung und hohe Verfügbarkeit für missionskritische Anwendungen	Sun Cluster-Software Sun Cluster Agents Sun Cluster Geographic Edition

TABELLE 1–4 In Sun Java Suites enthaltene Komponenten *(Fortsetzung)*

Suite	Geschäftsanforderung	Inhalt
Sun Java Communications Suite[1]	Sichere und verlässliche Messaging- und Zusammenarbeitsdienste	Access Manager Anwendungsserver Calendar Server* Communications Express* Delegated Administrator* Directory Server HADB Instant Messaging* Java DB Message Queue Messaging Server* Monitoring Console Web Proxy Server Web Server
Sun Java Identity Management Suite	Benutzer-Identitätsverwaltung innerhalb von Computer-Infrastrukturen und Anwendungsumgebungen	Access Manager Anwendungsserver Directory Server HADB Java DB Message Queue Monitoring Console Web Server

[1] Bei mit Asterisken (*) gekennzeichneten Kommunikationskomponenten handelt es sich um Komponenten, die in Java ES nicht mehr enthalten sind oder mithilfe des Java ES-Installationsprogramms installiert wurden. Diese Komponenten sind als Teil der Sun Java Communications Suite verfügbar.

TABELLE 1–4 In Sun Java Suites enthaltene Komponenten *(Fortsetzung)*

Suite	Geschäftsanforderung	Inhalt
Sun Java Web Infrastructure Suite	Webanwendungen und -dienste für kleine bis mittlere Unternehmen	Access Manager Anwendungsserver Directory Server HADB Java DB Message Queue Monitoring Console Service Registry Web Proxy Server Web Server

Grundlegende Hinweise zu Java ES

Beim Erstellen von auf Java ES-Software basierenden Unternehmenslösungen fallen einige Standardaufgaben an. Diese Aufgaben fallen je nach Startpunkt der Einführung von Java ES und der Art der Lösung, die erstellt und bereitgestellt werden soll, unterschiedlich aus.

Dieser Abschnitt behandelt zwei Aspekte des Arbeitens mit Java ES: Den Java ES-Lösungslebenszyklus und die verschiedenen Einführungsszenarios, die in der Regel bestehen.

Java ES-Lösungslebenszyklus

Die Aufgaben, die beim Erstellen von auf Java ES-Software basierenden Unternehmenslösungen anfallen, können in mehrere Phasen unterteilt werden. Diese sind in der folgenden Abbildung dargestellt. Die Abbildung zeigt außerdem, welche Kategorien von Java ES-Benutzern diese Aufgaben normalerweise durchführen.

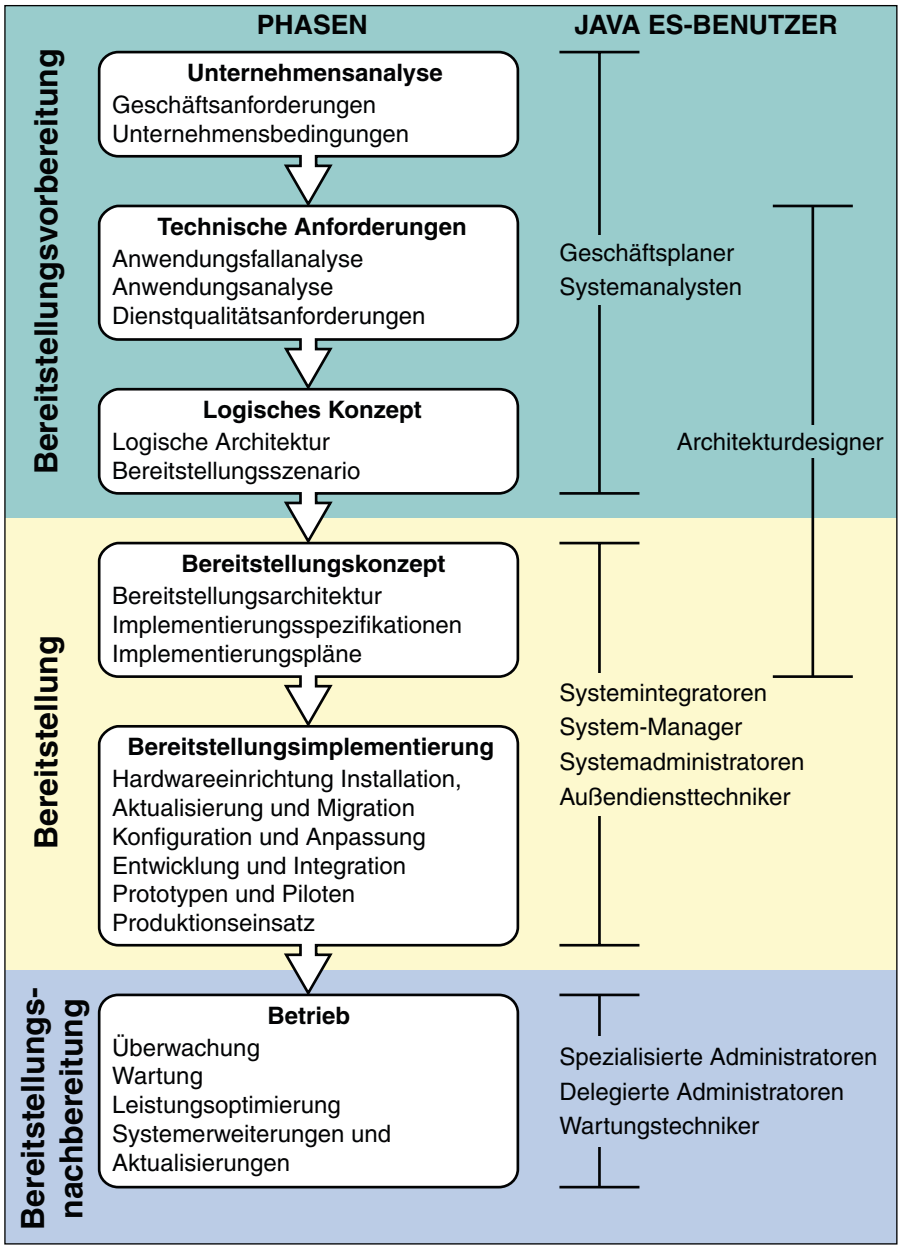

ABBILDUNG 1–3 Phasen des Lebenszyklus einer Lösung und Benutzerkategorien

Die in der vorherigen Abbildung gezeigten Lebenszyklusphasen können in die folgenden allgemeinen Gruppen unterteilt werden:

- **Bereitstellungsvorbereitung:** In diesen Phasen wird ein Geschäftsbedarf in ein Bereitstellungsszenario umgesetzt, bei dem es sich um eine logische Architektur und eine Qualitätsreihe von Dienstanforderungen handelt. Das Bereitstellungsszenario dient als Spezifikation für die Konzeption einer Bereitstellungsarchitektur.
- **Bereitstellung:** In dieser Phase wird ein Bereitstellungsszenario in eine Bereitstellungsarchitektur umgesetzt. Diese Architektur kann als Grundlage für die Genehmigung und Budgetierung des Projekts verwendet werden. Diese Bereitstellungsarchitektur ist auch die Grundlage für eine Implementierungsspezifikation, die die erforderlichen Details für die Bereitstellung (Erstellung, Testen und Einsatz) einer Softwarelösung in einer Produktionsumgebung enthält.
- **Bereitstellungsnachbereitung:** In der Einsatzphase läuft eine bereitgestellte Lösung unter Produktionsbedingungen und wird hinsichtlich der Leistung überwacht und optimiert. Die bereitgestellte Lösung wird bei Bedarf mit neuen Funktionen aktualisiert.

Die unter Abbildung 1–3 aufgeführten Aufgaben innerhalb der einzelnen Lebenszyklusphasen sind in Kapitel 4 genauer dargestellt.

Abbildung 1–3 zeigt die Java ES-Benutzer, die üblicherweise die Aufgaben in den einzelnen Lebenszyklusphasen durchführen. Die folgende Tabelle beschreibt die Kenntnisse und den Hintergrund der einzelnen Benutzerkategorien.

TABELLE 1–5 Java ES-Benutzerkategorien für Aufgaben im Lebenszyklus

Benutzer	Fähigkeiten und Kenntnisse	Phasen
Geschäftsplaner	Eher allgemeines, anstatt tiefer gehendes technisches Wissen.	Unternehmensanalyse
Systemanalyst	Verständnis der strategischen Orientierung des Unternehmens.	Technische Anforderungen
	Kenntnis der Geschäftsprozesse, -ziele und -anforderungen.	Logisches Konzept
Architekt	Sehr technisch orientiert.	Technische Anforderungen
	Breite Kenntnis der Bereitstellungsarchitekturen.	Logisches Konzept
	Kenntnis der neuesten Technologien.	Bereitstellungskonzept
	Verständnis der Geschäftsanforderungen und -beschränkungen.	
Systemintegrator	Sehr technisch orientiert.	Bereitstellungskonzept
Außendiensttechniker	Tiefgehende Kenntnis der IT-Umgebungen.	Bereitstellungsimplementierung
Systemadministrator	Erfahrung mit der Implementierung verteilter Softwarelösungen.	
System-Manager	Kenntnis der Netzwerkarchitektur, Protokolle, Geräte und Sicherheit.	
	Kenntnis der Skript- und Programmiersprachen.	

TABELLE 1–5 Java ES-Benutzerkategorien für Aufgaben im Lebenszyklus *(Fortsetzung)*

Benutzer	Fähigkeiten und Kenntnisse	Phasen
Spezialisierter Systemadministrator Delegated Administrator Wartungstechniker	Spezielle technische Kenntnisse oder Produktkenntnisse. Gute Kenntnis der Hardware, Plattformen, Verzeichnisse und Datenbanken. Geschult in der Überwachung, Fehlerbehebung und Aufrüstung von Software. Kenntnis der Systemverwaltung für Betriebssystemplattformen.	Betrieb

Java ES-Einführungsszenarios

Der Unternehmensbedarf, der zur Einführung von Java ES führt, ist sehr unterschiedlich. Das übergeordnete Ziel nahezu jeder Java ES-Bereitstellung stimmt jedoch mit einem der folgenden adoption scenarios überein:

- **Neues System:** Sie verfügen über kein Softwaresystem und beginnen mit der Bereitstellung der Java ES-Software, um eine neue Geschäftslösung zu unterstützen.
- **Verbesserung:** Sie verfügen über eine IT-Infrastruktur und ersetzen einige oder alle Teile Ihres Systems durch Java ES-Software. Üblicherweise werden Systeme oder Subsysteme ersetzt, weil deren Erhaltung zu kompliziert, zu eingeschränkt oder zu teuer wäre. Sie können beispielsweise bessere Sicherheit, höhere Verfügbarkeit, mehr Skalierbarkeit, mehr Flexibilität, weniger Komplexität, zusätzliche Funktionen (wie Single Sign-On) oder eine bessere Nutzung von IT-Ressourcen benötigen.
- **Erweiterung:**Sie verfügen über eine IT-Infrastruktur und stellen Java ES-Software bereit, die noch nicht Bestandteil Ihres Systems ist. Üblicherweise werden Softwaresysteme erweitert, weil neue Geschäftsanforderungen erfüllt werden müssen. Sie benötigen gegebenenfalls neue Funktionen, wie personalisierte Aggregation vorhandener Dienste über ein Java ES-Portal oder Java-Authentifizierung und -Autorisierung für vorhandene Dienste.
- **Aufrüstung:** Sie verfügen über eine IT-Infrastruktur bestehend aus einer früheren Version von Java ES oder Sun-Produkten, die Java ES vorausgegangen sind, und nehmen eine Aufrüstung auf die aktuelle Version der Java ES-Komponenten vor.

Jedes Einführungsszenario bietet eigene Überlegungen und Herausforderungen. Abhängig von Ihrem Einführungsszenario sind die in der Lebenszyklusphase zu lösenden Probleme und die zu investierenden Ressourcen unterschiedlich, wie in Abbildung 1–3 beschrieben.

Die folgenden Punkte gelten abgestuft für die Einführungsszenarios:

- **Migration:** Die Verbesserung oder Aufrüstung der vorhandenen Infrastruktur durch neue Software macht häufig die Migration von Daten aus vorhandenen in neue Systeme notwendig. Bei den Daten kann es sich um Konfigurations-, Benutzer- oder Anwendungsinformationen handeln. Gegebenenfalls müssen Sie aufgrund neuer Programmierschnittstellen außerdem die Geschäfts- oder Darstellungslogik migrieren.

- **Integration:** Das Hinzufügen neuer Software zu einem vorhandenen System oder das Ersetzen von Software-Subsystemen macht häufig eine Integration der neuen Softwarekomponenten in die verbleibenden Subsysteme erforderlich. Zur Integration kann die Entwicklung neuer Schnittstellenebenen, der Einsatz von J2EE Connectoren oder Ressourcen-Adaptern, die Neukonfiguration vorhandener Komponenten sowie die Implementierung von Datentransformationsschemata gehören.
- **Schulung:** Nahezu jede Veränderung der Infrastruktur zieht Änderungen der IT-Verfahren und der erforderlichen Kenntnisse nach sich. Ihre IT-Abteilung benötigt einen angemessenen Zeitraum, um die Kenntnisse für die neuen Java ES-Technologien zu erlangen oder die vorhandenen Kenntnisse weiterzuvermitteln.
- **Hardware:** Wenn Sie ein vorhandenes System oder Subsystem ersetzen, können es die Unternehmensbedingungen erforderlich machen, dass vorhandene Hardware weiterverwendet wird. Abhängig von Ihrem Einführungsszenario können die Hardwareressourcen zu einem wichtigen Faktor werden.

Die folgende Tabelle fasst die Art der Punkte zusammen, die bei den einzelnen Einführungsszenarios von Java ES von Bedeutung sind.

TABELLE 1–6 Aspekte unterschiedlicher Java ES-Einführungsszenarios

Einführungsszenario	Migration	Integration	Schulung	Hardware
Neues System	Nicht zutreffend	Relativ einfache Integration neuer Komponenten	Kann ein wichtiger Aspekt sein	Ausgleich zwischen Ausrüstungskosten und Arbeitskosten[1]
Verbesserung	Kann ein bedeutender Aspekt sein	Neue Komponenten müssen in vorhandene Systeme integriert werden	Kann ein wichtiger Aspekt sein	Kann wegen vorhandener Anlagen zu signifikanten Einschränkungen führen
Erweiterung	Üblicherweise kein Aspekt	Gegebenenfalls müssen neue Komponenten in vorhandene Systeme integriert werden	Gegebenenfalls ein wichtiger Aspekt	Macht üblicherweise neue Hardware notwendig, mit denselben Abwägungen wie bei einem neuen System
Aufrüstung	Kann ein wichtiger Aspekt sein	Relativ einfache Integration aufgerüsteter Komponenten	Relativ unwesentlicher Aspekt	Relativ unwesentlicher Aspekt

[1] Der Einsatz einiger leistungsstarker Computer erhöht im Allgemeinen die Anlagekosten, wenn weniger IT-Ressourcen benötigt werden. Der Einsatz vieler kleiner Computer senkt im Allgemeinen die Anlagekosten, wenn mehr IT-Ressourcen benötigt werden.

In diesem Kapitel enthaltene Schlüsselbegriffe

In diesem Abschnitt werden die wichtigsten in diesem Kapitel verwendeten technischen Bedingungen erläutert, wobei auf die Verwendung dieser Bedingungen im Zusammenhang mit Java ES besonders eingegangen wird.

Einführungsszenario Der wesentliche Grund für die Bereitstellung von Java ES-Software, durch die die Ausgangssituation hinsichtlich der Systemsoftware und das zu erreichende Ziel beschrieben wird. Die folgenden vier Einführungsszenarios sind in Java ES vorhanden: neues System, Ersatz, Erweiterung und Aufrüstung.

Komponente Eine Softwarelogikeinheit, die zum Aufbau von verteilten Anwendungen verwendet wird. Bei einer Komponente kann es sich um eine der in Java ES enthaltenen Systemkomponenten handeln oder um eine kundenspezifische entwickelte Anwendungskomponente. Eine Anwendungskomponente entspricht normalerweise einem verteilten Komponentenmodell (z. B. CORBA und der J2EE™-Plattform) und führt eine bestimmte Computerfunktion durch. Diese Komponenten bieten einzeln oder kombiniert business services und können als web services zusammengefasst werden.

Verteilte Unternehmensanwendung Eine Anwendung mit einer Logik, die sich über eine Netzwerk- oder Internetumgebung hinweg erstreckt (der Verteilungsaspekt) und mit einem Umfang und einer Größe, die die Anforderungen einer Produktionsumgebung oder eines Dienstanbieters erfüllen (der Unternehmensaspekt).

Endbenutzer Eine Person, die eine verteilte Anwendung nutzt, häufig über eine grafische Benutzeroberfläche, wie die Oberfläche eines Internetbrowsers oder eines mobilen Geräts. Die Anzahl der gleichzeitig von einer Anwendung unterstützten Endbenutzer bildet einen wichtigen Faktor für die Bereitstellungs- architektur der Anwendung.

Dienst Eine Softwarefunktion, die von einem oder mehreren clients ausgeführt wird. Diese Funktion kann auf einer sehr niedrigen Ebene, beispielsweise eine Arbeitsspeicherverwaltung, oder auf hoher Ebene, beispielsweise eine Kreditüberprüfung durch einen Geschäftsdienst, dargestellt werden. Ein Dienst auf hoher Ebene kann aus einer Reihe einzelner Dienste bestehen. Die Dienste können lokal (für lokale Clients verfügbar) oder verteilt (für Remote-Clients verfügbar) sein.

Produktkomponente Java ES Systemdienstkomponenten, durch die die Hauptinfrastrukturdienste von Java ES bereitgestellt werden und Java ES-Dienstqualitätskomponenten , durch die solche Systemdienste verbessert werden. Produktkomponenten können innerhalb des Java ES-Installationsprogramms ausgewählt werden.

Dienstqualitätskomponente Eine Java ES enthaltene Systemkomponente. Diese Komponenten verbessern die Verfügbarkeit, Sicherheit, Skalierbarkeit, Zweckmäßigkeit sowie andere Qualitäten der Systemdienstkomponenten und verteilten Anwendungskomponenten.

Gemeinsam genutzte Komponente Eine in Java ES enthaltene Systemkomponente. Gemeinsam genutzte Komponenten, normalerweise Bibliotheken, stellen für andere Systemkomponenten lokale Dienste bereit.

Systemkomponente Alle Softwarepakete oder Gruppen von Paketen, die Teil von Java ES sind und vom Java ES-Installationsprogramm installiert werden. Die folgenden Arten von Systemkomponenten sind vorhanden: Produktkomponenten, die Java ES-Infrastrukturdienste bereitstellen und gemeinsam genutzte Komponenten, die lokale Dienste für andere Systemkomponenten bereitstellen.

Systemdienst Einer oder mehrere verteilte services, die die von einem Java ES bereitgestellten einzigartigen Funktionen definieren. Systemdienste erfordern in der Regel die Unterstützung durch eine Reihe von Dienstqualitätskomponenten, gemeinsam genutzten Komponenten oder mehrere von beiden Komponentenarten.

Systemdienstkomponente Eine in Java ES enthaltene Systemkomponente. Systemdienstkomponenten bilden die wesentlichen Java ES-Infrastrukturdienste von: Zugangsdiensten, Identitäts- und Sicherheitsdiensten, Web- und Anwendungsdiensten, sowie Verfügbarkeitsdiensten.

◆ ◆ ◆ KAPITEL 2

Java ES-Lösungsarchitekturen

Dieses Kapitel bietet eine Übersicht über die Architekturkonzepte, auf denen die Java ES-Lösungen basieren. In diesem Kapitel wird gezeigt, wie Systemdienstkomponenten und Dienstqualitätskomponenten eingesetzt werden, um verteilte Unternehmenslösungen zu unterstützen.

Architekturen von Java ES-Lösungen bieten zwei Aspekte: Eine Logische Architektur und eine Bereitstellungs- architektur. Die logische Architektur stellt die Interaktionen zwischen den logischen Modulblöcken (den Softwarekomponenten) einer Lösung dar. Die Bereitstellungsarchitektur stellt die Zuordnung der logischen Architektur zu einer physischen Computerumgebung dar. Java ES-Komponenten spielen sowohl in logischen Architekturen als auch in Bereitstellungsarchitekturen eine wichtige Rolle.

In diesem Kapitel werden ein Architektur-Framework für die Konzeption der Architektur von Java ES-Lösungen sowie eine Beispiellösung, die auf diesem Framework aufbaut, beschrieben. Das Kapitel enthält die folgenden Abschnitte:

Java ES-Architektur-Framework

Java ES-Komponenten unterstützen die Bereitstellung verteilter Softwarelösungen. Damit die benötigte Funktionalität auf der von den Geschäftsanforderungen vorgegebenen Ebene der Leistung, Verfügbarkeit, Sicherheit, Skalierbarkeit und Zweckmäßigkeit erreicht wird, müssen solche Softwarelösungen korrekt entworfen werden.

Bei der Konzeption von Lösungen in Unternehmensstärke sind verschiedene Architekturdimensionen zu berücksichtigen. Diese Dimensionen repräsentieren unterschiedliche Perspektiven, aus denen die Interaktionen der vielen, solche Systeme

bildenden Softwarekomponenten betrachtet werden. Insbesondere bei der Konzeption verteilter Systeme sind die drei folgenden Architekturdimensionen zu berücksichtigen:

- **Infrastrukturdienst-Abhängigkeiten.** Diese Dimension hebt die Rolle von Systemdienstkomponenten bei der Unterstützung verteilter Lösungen hervor (siehe „Systemdienst- komponenten" auf Seite 20).
- **Logische Schichten.** Diese Dimension stellt die logische und physische Unabhängigkeit von Lösungskomponenten in den Vordergrund, die in einer Netzwerk- oder Internetumgebung bereitgestellt werden sollen.
- **Dienstqualität.** Diese Dimension stellt dar, wie Dienstqualitätsanforderungen (Verfügbarkeit, Sicherheit, Skalierbarkeit und Zweckmäßigkeit) erreicht werden, wobei die Rolle der Dienstqualitätskomponenten berücksichtigt wird (siehe „Dienstqualitätskomponenten" auf Seite 22).

In der folgenden Abbildung sind diese drei Dimensionen der Lösungsarchitektur dargestellt.

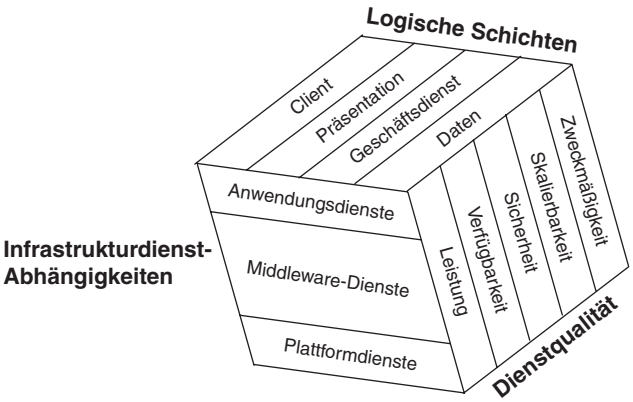

ABBILDUNG 2–1 Dimensionen einer Java ES-Lösungsarchitektur

Diese drei Dimensionen bilden zusammen ein einziges Framework, das die Beziehungen zwischen den Softwarekomponenten (sowohl application components als auch Infrastrukturkomponenten) berücksichtigt, die für das Erreichen der für eine Softwarelösung benötigten Dienstfunktionen und der Dienstqualität erforderlich sind.

In den folgenden Abschnitten werden die drei Dimensionen nacheinander beschrieben. Anschließend wird eine Synthese der drei Dimensionen in einem zusammengeführten Framework vorgenommen.

Dimension 1: Infrastrukturdienstabhängigkeiten

Die interagierenden Softwarekomponenten verteilter Unternehmensanwendungen setzen Basisinfrastrukturdienste voraus, die den verteilten Komponenten die Kommunikation

untereinander, die Koordination ihrer Arbeit, die Implementierung eines sicheren Zugriffs usw. ermöglichen. In diesem Abschnitt wird die Schlüsselrolle einiger Java ES-Komponenten bei der Bereitstellung dieser Infrastrukturdienste erläutert.

Infrastrukturdienstebenen

Bei der Konzeption eines verteilten Softwaresystems müssen Sie unabhängig davon, ob das Softwaresystem überwiegend aus selbst entwickelten oder fertigen Java ES-Komponenten besteht, einige Infrastrukturdienste berücksichtigen. Diese Dienste arbeiten auf vielen Ebenen.

Die Infrastrukturdienstabhängigkeiten der Lösungsarchitektur werden in Abbildung 2–2 dargestellt. Die in dieser Abbildung dargestellten Ebenen sind eine vergrößerte Ansicht der in Abbildung 1–1 gezeigten Infrastrukturdienstebene. Die Hierarchie der in Abbildung 2–2 dargestellten Dienste und die Abhängigkeiten zwischen den Diensten bilden eine wichtige Dimension der logischen Architektur der Lösung. Diese Infrastrukturdienste liefern das Grundprinzip für Java ES-Systemdienstkomponenten (siehe „Systemdienst- komponenten” auf Seite 20).

Im Allgemeinen können die in der folgenden Abbildung dargestellten Dienste in drei große Gruppen unterteilt werden: Plattformdienste auf niedrigster Ebene, Anwendungsdienste auf höchster Ebene und eine Gruppe von Middleware-Diensten, die ihren Namen ihrer Position zwischen den beiden anderen Gruppen verdanken.

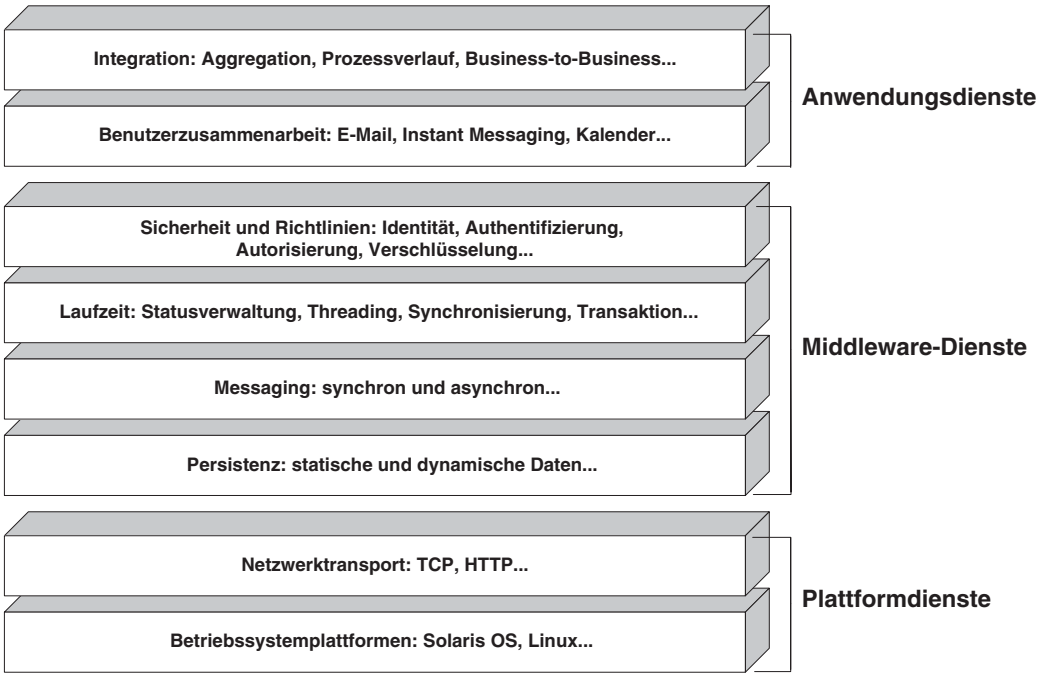

ABBILDUNG 2–2 Dimension 1: Infrastrukturdienstebenen

Die folgenden Beschreibungen der verschiedenen Infrastrukturdienstebenen beziehen sich, wenn relevant, auf Java-Programmiersprachenprodukte und werden von der niedrigsten bis zur höchsten Ebene aufgeführt, wie in Abbildung 2–2 dargestellt:

- **Betriebssystemplattformen.** Bieten die Basisunterstützung für alle Prozesse, die auf einem Computer ausgeführt werden. Das Betriebssystem verwaltet die physischen Geräte sowie den Arbeitsspeicher, die Threads und andere Ressourcen, die zur Unterstützung der Java Virtual Machine (JVM™-Machine) erforderlich sind.
- **Netzwerktransport.** Bietet die grundlegende Netzwerkunterstützung für die Kommunikation zwischen den verteilten Anwendungskomponenten, die auf den verschiedenen Computern ausgeführt werden. Zu diesen Diensten gehört auch die Unterstützung von Protokollen wie TCP und HTTP. Andere Kommunikationsprotokolle auf höheren Ebenen (siehe Messaging-Ebene) sind von diesen grundlegenden Transportdiensten abhängig.
- **Persistenz.** Bietet Unterstützung für den Zugriff auf und die Speicherung von statischen Daten (wie Benutzer-, Verzeichnis- oder Konfigurationsinformationen) und dynamischen Anwendungsdaten (Informationen, die häufig aktualisiert werden).
- **Messaging:** Bietet Unterstützung für die synchrone und asynchrone Kommunikation zwischen den Anwendungskomponenten. Synchrones Messaging besteht im Senden und Empfangen von Nachrichten in Echtzeit und umfasst Remote-Methodenaufrufe (RMI)

zwischen J2EE-Komponenten und SOAP-Interaktionen zwischen Webdiensten. Asynchrones Messaging besteht in einer Kommunikation, bei der das Senden einer Nachricht nicht davon abhängt, ob der Konsument bereit ist, diese sofort zu empfangen. Die Spezifikationen für asynchrones Messaging, beispielsweise Java Message Service (JMS) und ebXML, sorgen für garantierte Zuverlässigkeit und andere Aspekte der Messaging-Semantik.

- **Laufzeit:** Bietet die Unterstützung, die für jedes verteilte Komponentenmodell, beispielsweise J2EE- oder CORBA-Modelle, erforderlich ist. Neben dem Remote-Methodenaufruf, der für eng miteinander verknüpfte verteilte Komponenten benötigt wird, umfassen die Laufzeitdienste die Komponentenstatusverwaltung (Lebenszyklusverwaltung), die Thread-Pool-Verwaltung, die Synchronisierung (Mutex-Sperrung), Persistenzdienste, die verteilte Transaktionsüberwachung und die verteilte Ausnahmenverarbeitung. In einer J2EE-Umgebung werden diese Laufzeitdienste von EJB-Containern, Webcontainern und nachrichtengesteuerten Bean-Containern (MDB-Containern) auf einem Anwendungs- oder Webserver bereitgestellt.
- **Sicherheit und Richtlinie.** Bietet Unterstützung für den sicheren Zugriff auf Anwendungsressourcen. Diese Dienste umfassen die Unterstützung für Richtlinien, die den gruppen- oder rollenbasierten Zugriff auf verteilte Ressourcen steuern, sowie Single Sign-On-Funktionen. Single Sign-On ermöglicht, dass die Authentifizierung eines Benutzers bei einem Dienst in einem verteilten System automatisch auf andere Dienste (J2EE-Komponenten, Geschäftsdienste und Webdienste) in diesem System angewendet wird.
- **Benutzerzusammenarbeit:** Bietet Dienste, die bei der Unterstützung der direkten Kommunikation zwischen den Benutzern und der Zusammenarbeit der Benutzer in Unternehmens- und Internetumgebungen eine wichtige Rolle spielen. Diese Dienste sind Geschäftsdienste auf Anwendungsebene, die in der Regel von eigenständigen Servern, wie beispielsweise von einem E-Mail- oder einem Calendar Server, bereitgestellt werden.
- **Integration:** Stellt die Dienste für die Aggregation vorhandener Geschäftsdienste bereit. Bietet eine gemeinsame Schnittstelle für den Zugriff auf die Dienste, wie dies in einem Portal der Fall ist, indem die Dienste über eine Prozess-Engine integriert werden, die diese innerhalb eines Produktionsworkflows koordiniert. Die Integration kann auch in Form von Business-to-Business-Interaktionen zwischen verschiedenen Unternehmen erfolgen.

Die in Abbildung 2–2 dargestellten Dienstebenen spiegeln die gegenseitige Abhängigkeit der Infrastrukturdienste wider, von der untersten Ebene der Betriebssystemdienste bis hinauf zur höchsten Ebene der Anwendungs- und Integrationsdienste. Im Allgemeinen ist jeder Dienst von untergeordneten Diensten abhängig und unterstützt selbst übergeordnete Dienste. Abbildung 2–2 stellt jedoch keine strenge Ebenenschichtung der Infrastrukturdienste dar. Dienste höherer Ebenen können direkt mit Diensten niedrigerer Ebenen interagieren, ohne von Zwischenebenen abhängig zu sein. So sind beispielsweise bestimmte Laufzeitdienste direkt von den Plattformdiensten abhängig, ohne eine der dazwischen liegenden Dienstebenen zu benötigen. Zusätzlich könnten auch andere Dienstebenen, wie Überwachungs- oder Verwaltungsdienste, ebenfalls in diese Konzeptdarstellung aufgenommen werden.

Java ES-Infrastrukturdienstkomponenten

Java ES-Komponenten implementieren die verteilten Infrastrukturdienstebenen, die in Abbildung 2–2 dargestellt sind. Die Positionierung der Systemdienstkomponenten innerhalb der verschiedenen Ebenen wird in der folgenden Abbildung dargestellt:

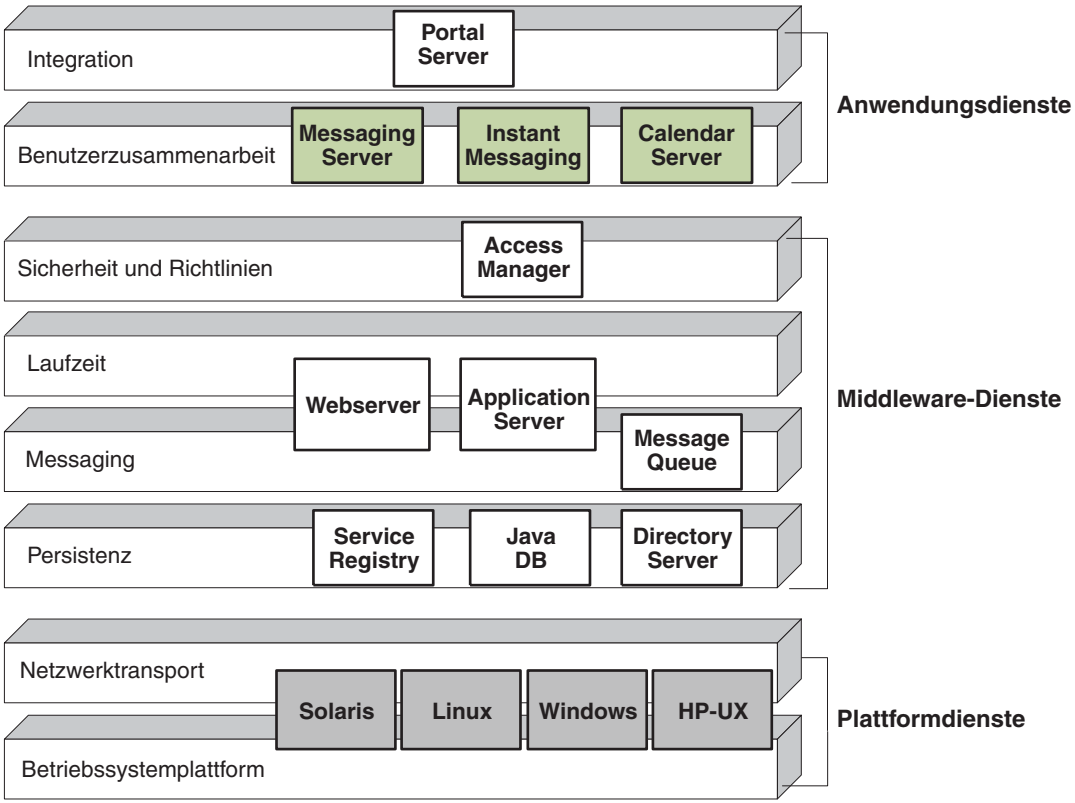

ABBILDUNG 2–3 Java ES-Systemdienstkomponenten

Hinweis – Die in der Abbildung schattiert dargestellten Kästchen verweisen auf Komponenten, die nicht in Java ES enthalten sind. Zwar sind Benutzerzusammenarbeitskomponenten nicht Bestandteil von Java ES, sie werden jedoch oftmals gemeinsam mit Java ES-Komponenten bereitgestellt und innerhalb von Java ES-Architekturen verwendet. Diese Komponenten sind Teil der Sun Java Communications Suite und werden in diesem Dokument nur zu Darstellungszwecken referenziert. Betriebssystemplattformen sind darüber hinaus formell betrachtet nicht Teil von Java ES, sie sind jedoch in der Abbildung aufgeführt, um die Betriebssystemplattformen aufzuzeigen, auf denen Java ES-Komponenten unterstützt werden.

Java ES-Infrastrukturdienstabhängigkeiten

Im Allgemeinen ist jede der in Abbildung 2–3 dargestellten Java ES-Systemdienstkomponenten von in der Infrastruktur unter ihr liegenden Komponenten anhängig und unterstützt darüber liegende Komponenten. Diese Beziehung von Abhängigkeit und Unterstützung bildet einen Schlüsselfaktor bei der Konzeption logischer Architekturen.

Die folgende Abbildung zeigt die spezifischen Abhängigkeiten zwischen den Java ES-Systemdienstkomponenten, die in Abbildung 2–3 von oben nach unten verlaufend dargestellt werden.

TABELLE 2–1 Beziehungen zwischen Java ES-Systemdienstkomponenten

Komponente	Abhängig von	Bietet Unterstützung für
Portal Server	Anwendungsserver oder Web Server Access Manager Directory Server Wenn zur Verwendung von entsprechenden Kanälen konfiguriert: Calendar Server, Messaging Server und Instant Messaging[1]	Keine
Access Manager	Anwendungsserver oder Web Server Directory Server	Portal Server Wenn konfiguriert für Single Sign-On: Calendar Server, Messaging Server und Instant Messaging
Anwendungsserver	Message Queue Directory Server (für verwaltete Objekte)	Portal Server Access Manager
Message Queue	Directory Server (für verwaltete Objekte)	Anwendungsserver
Web Server	Access Manager (für die Zugriffssteuerung)	Portal Server Access Manager
Directory Server	Keine	Portal Server Access Manager Calendar Server Messaging Server Instant Messaging

[1] Calendar Server, Messaging Server und Instant Messaging-Komponenten sind als Teil der Sun Java Communications Suite verfügbar.

TABELLE 2–1 Beziehungen zwischen Java ES-Systemdienstkomponenten *(Fortsetzung)*

Komponente	Abhängig von	Bietet Unterstützung für
Service Registry	Java-DB	Application Server-basierte Komponenten
Java-DB	Keine	Service Registry

Dimension 2: Logische Schichten

Die interagierenden Softwarekomponenten verteilter Unternehmensanwendungen können logischen Schichten zugeordnet werden. Diese Schichten stellen die logische und physische Unabhängigkeit von Softwarekomponenten auf der Grundlage der Art der von ihnen angebotenen Dienste dar.

Die Dimension der logischen Schichten in einer Lösungsarchitektur ist in der folgenden Abbildung dargestellt.

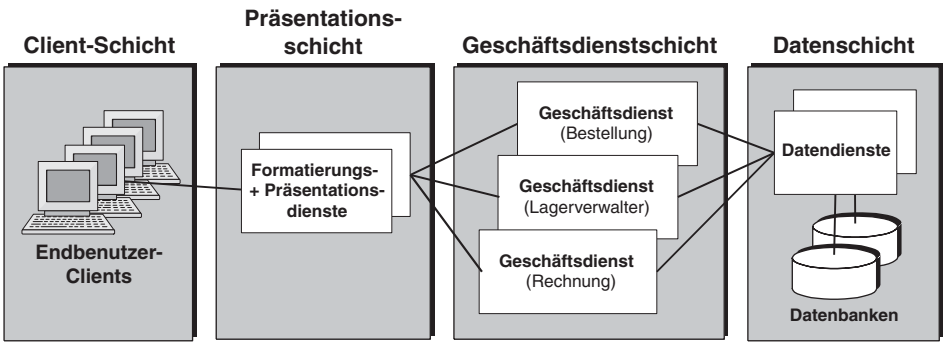

ABBILDUNG 2–4 Dimension 2: Logische Schichten für verteilte Unternehmensanwendungen

Logische Schichtenarchitekturen stellen überwiegend die in Abbildung 1–1 dargestellte Ebene der verteilten Unternehmensanwendung dar. Die unter „Infrastrukturdienstebenen" auf Seite 35 behandelten Java ES-Systemdienstkomponenten unterstützen Anwendungskomponenten aller in Abbildung 2–4 dargestellten logischen Schichten. Während sich logische Schichtenkonzepte vorwiegend auf benutzerdefinierte Unternehmensanwendungen beziehen, beziehen sie sich darüber hinaus auf Dienste, die von Komponenten der Sun Java Communications Suite und einigen Zugangsdiensten bereitgestellt werden.

Beschreibung der logischen Schichten

Dieser Abschnitt enthält Kurzbeschreibungen der vier logischen Schichten, die in Abbildung 2–4 dargestellt sind. Die Beschreibungen beziehen sich auf Anwendungskomponenten, die

mithilfe eines auf der J2EE-Plattform basierenden Komponentenmodells implementiert wurden. Andere verteilte Komponentenmodelle hingegen, beispielsweise CORBA, unterstützen ebenfalls diese Architektur.

- **Client-Schicht:** Die Client-Schicht umfasst die Anwendungslogik, auf die ein Endbenutzer über eine Benutzeroberfläche direkt zugreift. Die Logik in der Client-Schicht kann browserbasierte Clients, auf einem Desktop-Computer ausgeführte Java-Komponenten oder auf einem Handheld-Gerät ausgeführte mobileJava™ 2 Platform Micro Edition-Clients (J2ME™-Plattform) enthalten.
- **Präsentationsschicht:** Die Präsentationsschicht umfasst Anwendungslogik, die Daten für die Zustellung an die Client-Schicht vorbereitet und Anforderungen der Client-Schicht für die Zustellung an die Back-End-Geschäftslogik verarbeitet. Die Logik in der Präsentationsschicht umfasst in der Regel J2EE-Komponenten, wie Java Servlet-Komponenten oder JSP-Komponenten, die Daten für die Zustellung im HTML- oder XML-Format vorbereiten oder Verarbeitungsanforderungen empfangen. Diese Schicht kann auch einen Zugangsdienst enthalten, der einen personalisierten, sicheren und benutzerdefinierten Zugriff auf die in der Geschäftsdienstschicht vorhandenen business services ermöglichen kann.
- **Geschäftsdienstschicht:** Die Geschäftsdienstschicht umfasst die Logik, die die Hauptfunktionen der Anwendung ausführt: Datenverarbeitung, Implementierung der Geschäftsregeln, Koordination mehrerer Benutzer und Verwaltung externer Ressourcen, wie Datenbanken oder Legacy-Systeme. In der Regel umfasst diese Schicht eng miteinander verknüpfte Komponenten, die dem verteilten J2EE-Komponentenmodell entsprechen, wie Java-Objekte, EJB-Komponenten oder nachrichtengesteuerte Beans. Einzelne J2EE-Komponenten können für die Bereitstellung komplexer Geschäftsdienste, beispielsweise Inventardienste oder Steuerberechnungsdienste, zusammengefügt werden. Einzelne Komponenten und Dienstgruppen können in einem dienstorientierten Architekturmodell zu lose miteinander verknüpften Webdiensten zusammengefasst werden, die dem Schnittstellenstandard SOAP (Simple Object Access Protocol) entsprechen. Geschäftsdienste können auch als eigenständige Server, beispielsweise als Unternehmens-Calendar Server oder Messaging Server, erstellt werden.
- **Datenschicht:** Die Datenschicht umfasst Dienste, durch die persistente Daten bereitgestellt werden, die von der Geschäftslogik verwendet werden. Die Daten können Anwendungsdaten sein, die in einem Datenbankverwaltungssystem gespeichert sind, oder Ressourcen- und Verzeichnisinformationen, die in einem Lightweight Directory Access Protocol-Datenspeicher (LDAP-Datenspeicher) gespeichert sind. Die Datendiensten können auch Daten aus externen Quellen oder Daten aus Legacy-Computersystemen enthalten.

Logische und physische Unabhängigkeit

Die in Abbildung 2–4 gezeigte Architekturdimension betrachtet vor allem die logische und physische Unabhängigkeit von Komponenten, die durch vier separate Schichten dargestellt

wird. Diese Schichten stellen die Partitionierung der Anwendungslogik über verschiedene in einer Netzwerkumgebung vorhandene Computer hinweg dar:

- **Logische Unabhängigkeit:** Die vier Schichten des Architekturmodells stellen die logische Unabhängigkeit dar: Sie können die Anwendungslogik in einer Schicht (beispielsweise in der Geschäftsdienstschicht) unabhängig von der Logik in den anderen Schichten ändern. Genauso können Sie Ihre Implementierung der Geschäftslogik ändern, ohne die Logik in der Präsentations- oder Client-Schicht ändern oder aktualisieren zu müssen. Diese Unabhängigkeit bedeutet, dass Sie beispielsweise neue Typen von Client-Komponenten aufnehmen können, ohne die Geschäftsdienstkomponenten zu ändern.
- **Physische Unabhängigkeit:** Die vier Schichten stellen auch die physische Unabhängigkeit dar: Sie können die Logik in vier verschiedenen Schichten auf verschiedenen Hardwareplattformen (verschiedene CPU-Konfigurationen, Chipsätze und Betriebssysteme) bereitstellen. Diese Unabhängigkeit ermöglicht es, verteilte Anwendungskomponenten auf den Computern auszuführen, die den einzelnen Computeranforderungen am ehesten entsprechen und am besten zur Maximierung der Netzwerkbandbreite geeignet sind.

  Wie Sie Anwendungs- oder Infrastrukturkomponenten auf eine Hardwareumgebung (Ihre Bereitstellungsarchitektur) abbilden, hängt von vielen Faktoren ab, die durch die Größe und Komplexität Ihrer Softwarelösung bestimmt werden. Bei sehr kleinen Bereitstellungen kann eine Bereitstellungsarchitektur nur wenige Computer betreffen. Bei großen Bereitstellungen kann die Zuordnung von Komponenten zu einer Hardwareumgebung Faktoren wie Geschwindigkeit und Leistung einzelner Computer, Geschwindigkeit und Bandbreite von Netzwerkleitungen, Sicherheits- und Firewall-Aspekte sowie Replikationsstrategien für Komponenten berücksichtigen.

Auf Systemkomponenten angewendete geschichtete Architektur

Wie in Abbildung 2–3 gezeigt, bieten Java ES-Infrastrukturdienstkomponenten die zugrundeliegende Infrastrukturunterstützung für verteilte Softwarelösungen. Einige dieser Lösungen beinhalten Dienste auf Anwendungsebene, die von Komponenten der Sun Java Communications Suite und einigen Java ES-Komponenten bereitgestellt werden. Diese Lösungen verwenden den konzeptionellen Ansatz der logischen Schichten.

Die von Messaging Server bereitgestellten E-Mail-Kommunikationsdienste sind so implementiert, dass sie bestimmte logisch gesonderte Konfigurationen von Messaging Server verwenden. Diese gesonderten Konfigurationen bieten jeweils einen gesonderten Satz von Diensten. Bei der Konzeption von Messaging-Lösungen werden diese gesonderten Konfigurationen in Form von separaten Komponenten dargestellt, die sich in verschiedenen logischen Schichten befinden, wie in der folgenden Abbildung dargestellt, in der die Verbindungslinien von Komponenten Interaktionen darstellen.

Hinweis – In der folgenden Abbildung soll keine vollständige logische Architektur dargestellt werden. Zur vereinfachten Darstellung wurden einige Java ES-Komponenten ausgelassen.

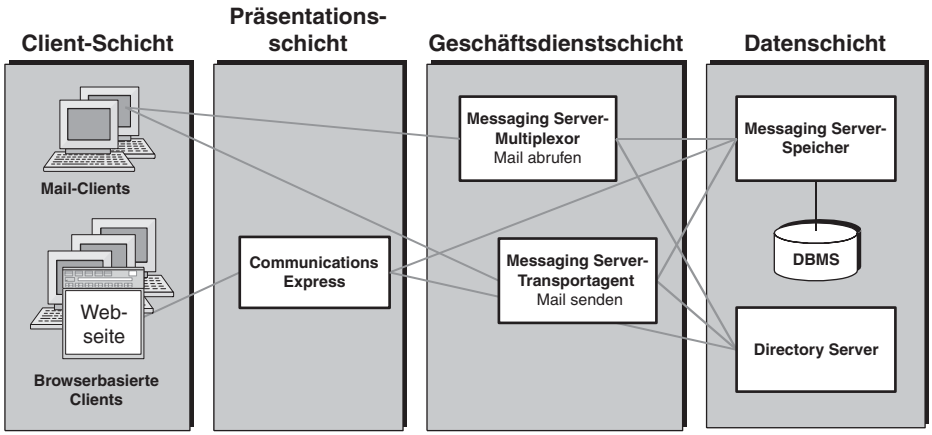

ABBILDUNG 2–5 Messaging Server : Beispiel für eine Schichtenarchitektur

Hinweis – Zwar sind Kommunikationskomponenten nicht Teil von Java ES, sie werden jedoch häufig mit Java ES-Komponenten und innerhalb von Java ES-Architekturen verwendet. Diese Kommunikationskomponenten sind Teil der Sun Java Communications Suite und werden in diesem Dokument nur zu Darstellungszwecken referenziert.

Die logische Trennung von Messaging Server -Funktionen in verschiedene Schichten erlaubt logisch unterschiedliche Konfigurationen von Messaging Server , die auf verschiedenen Computern in einer physischen Umgebung bereitgestellt werden. Durch die physische Trennung wird eine größere Flexibilität bei der Einhaltung der Dienstqualitätsanforderungen ermöglicht (siehe„Dimension 3: Dienstqualität" auf Seite 43). Dies bietet beispielsweise unterschiedliche Verfügbarkeitslösungen für unterschiedliche Instanzen und unterschiedliche Sicherheitsimplementierungen für unterschiedliche Messaging Server -Funktionen.

Dimension 3: Dienstqualität

Die beiden vorherigen Architekturdimensionen (Infrastrukturdienstabhängigkeiten und logische Schichten) betreffen überwiegend die logischen Aspekte der Architektur, nämlich, welche Komponenten auf welche Weise für die Interaktion erforderlich sind, um Dienste für Endbenutzer bereitzustellen. Eine ebenso wichtige Dimension jeder bereitgestellten Lösung ist die Fähigkeit, Dienstqualitätsanforderungen zu erfüllen.

Die Dimension der Dienstqualität einer Lösungsarchitektur stellt die Rolle der Java ES-Dienstqualitätskomponenten in den Vordergrund.

Dienstqualitäten

Aufgrund der Tatsache, dass Internet- und E-Commerce-Dienste für Geschäftsvorgänge immer wichtiger werden, stellt die Leistung, Verfügbarkeit, Sicherheit, Skalierbarkeit und Zweckmäßigkeit dieser Dienste eine der wichtigsten Dienstqualitätsanforderungen umfangreicher, extrem leistungsstarker Bereitstellungsarchitekturen dar.

Für die Konzeption einer erfolgreichen Softwarelösung müssen Sie relevante Dienstqualitätsanforderungen aufstellen und eine Architektur entwerfen, die diese erfüllt. Um Dienstqualitätsanforderungen zu definieren, werden einige wichtige Dienstqualitäten verwendet. Die folgende Tabelle führt diese Dienstqualitäten auf.

TABELLE 2–2 Dienstqualitäten mit Auswirkung auf die Lösungsarchitektur

Systemdienstqualitäten	Beschreibung
Leistung	Die Messung der Antwortzeit und -latenz in Bezug auf die Benutzerladebedingungen.
Verfügbarkeit	Ein Maß dafür, wie oft die Ressourcen und Dienste eines Systems für Endbenutzer verfügbar sind (die *Betriebszeit* eines Systems).
Sicherheit	Eine komplexe Kombination von Faktoren, die die Integrität eines Systems und seiner Benutzer beschreibt. Zur Sicherheit gehören die physische Sicherheit der Systeme, die Netzwerksicherheit, die Anwendungs- und Datensicherheit (Authentifizierung und Autorisierung der Benutzer) sowie der sichere Transport von Informationen.
Skalierbarkeit	Die Möglichkeit, einem bereitgestellten System im Laufe der Zeit Kapazität hinzuzufügen. Skalierbarkeit beinhaltet üblicherweise das Hinzufügen von Ressourcen zum System, ohne dass Änderungen an der Bereitstellungsarchitektur vorgenommen werden müssen.
Latente Kapazität	Die Fähigkeit eines Systems, eine außergewöhnliche Spitzenauslastung ohne zusätzliche Ressourcen zu bewältigen.
Zweckmäßigkeit	Die Leichtigkeit, mit der ein bereitgestelltes System gewartet werden kann, einschließlich der Systemüberwachung, Problembehebung und Aktualisierung von Soft- und Hardwarekomponenten.

Die Dimension der Dienstqualität wirkt sich stark auf die Bereitstellungsarchitektur einer Lösung aus: Wie Anwendungskomponenten und Infrastrukturkomponenten in einer physischen Umgebung bereitgestellt werden.

Die Dienstqualitäten, die die Bereitstellungsarchitektur beeinflussen, sind eng miteinander verbunden: Anforderungen an eine Systemqualität wirken sich häufig auf die Konzeption der anderen Dienstqualitäten aus. So kann beispielsweise ein höheres Sicherheitsniveau die

Leistung beeinträchtigen, was wiederum die Verfügbarkeit beeinflusst. Das Hinzufügen weiterer Computer, um Verfügbarkeitsprobleme durch Redundanz zu beheben, wirkt sich häufig auf die Wartungskosten (Zweckmäßigkeit) aus.

Das Verständnis der Beziehungen zwischen den Dienstqualitäten und ihre Abstimmung ist eine wichtige Voraussetzung für die Konzeption von Architekturen, die sowohl die Geschäftsanforderungen als auch die Geschäftsbeschränkungen einhalten.

Java ES-Dienstqualitätskomponenten

Einige Java ES-Komponenten werden hauptsächlich verwendet, um die Dienstqualitäten von Systemdienstkomponenten oder verteilten Anwendungskomponenten zu verbessern. Diese Softwarekomponenten werden oft zusammen mit Hardwarekomponenten, wie Lastausgleichsmodulen und Firewalls, eingesetzt.

Die unter„Dienstqualitätskomponenten" auf Seite 22 vorgestellten Java ES-Dienstqualitätskomponenten können wie folgt zusammengefasst werden:

- **Verfügbarkeitskomponenten:** Bieten eine beinahe unterbrechungsfreie Betriebszeit für bereitgestellte Lösungen.
- **Zugangskomponenten:** Sorgen für einen sicheren Internetzugriff auf Systemdienste und bieten häufig auch eine Routing-Funktion.
- **Überwachungskomponenten:** Liefern Echtzeitinformationen zu Java ES-Komponenten

Die folgende Tabelle führt die wichtigsten Java ES-Dienstqualitätskomponenten aus der Perspektive der Architektur auf und zeigt, auf welche Systemqualitäten sie sich am meisten auswirken.

TABELLE 2–3 Dienstqualitätskomponenten und die beeinflussten Systemqualitäten

Komponente	Beeinflusste Systemqualitäten
High Availability Session Store	Verfügbarkeit
Monitoring Console	Zweckmäßigkeit
Portal Server, Secure Remote Access	Sicherheit Skalierbarkeit
Sun Cluster	Verfügbarkeit Skalierbarkeit
Sun Cluster Geographic Edition	Verfügbarkeit Skalierbarkeit

TABELLE 2–3 Dienstqualitätskomponenten und die beeinflussten Systemqualitäten *(Fortsetzung)*

Komponente	Beeinflusste Systemqualitäten
Web Proxy Server	Sicherheit
	Leistung
	Skalierbarkeit

Sun Cluster-Software

Sun Cluster-Software bietet Hochverfügbarkeits- und Skalierbarkeitsdienste für Java ES-Komponenten sowie für Anwendungen, die von der Java ES-Infrastruktur unterstützt werden. Ein Cluster besteht aus lose miteinander verbundenen Computern, die zusammen eine einzelne Client-Ansicht der Dienste, Systemressourcen und Daten bieten. Intern verwendet der Cluster redundante Computer, Interconnects, Datenspeicher und Netzwerkschnittstellen zur Bereitstellung der Hochverfügbarkeit für clusterbasierte Dienste und Daten.

Die Sun Cluster-Software überwacht permanent den Zustand der Mitgliedsknoten und anderer Cluster-Ressourcen. Im Fehlerfall greift die Sun Cluster-Software ein und löst das Failover der überwachten Ressourcen aus und nutzt dabei die interne Redundanz, um einen nahezu kontinuierlichen Zugriff auf diese Ressourcen zu realisieren.

Sun Cluster-Datendienstpakete (auch als Sun Cluster Agenten bezeichnet) sind für alle Java ES-Systemdienstkomponenten verfügbar. Sie können auch für selbst entwickelte Anwendungskomponenten Agenten schreiben.

Aufgrund der Steuerung durch die Sun Cluster-Software kann ein Cluster auch skalierbare Dienste bereitstellen. Durch Nutzung des globalen Dateisystems des Clusters und wegen der Fähigkeit, Infrastruktur- oder Anwendungsdienste auf mehreren Knoten in einem Cluster auszuführen, kann eine verstärkte Anforderung dieser Dienste auf mehrere Instanzen der Dienste aufgeteilt werden. Daher kann Sun Cluster-Software bei richtiger Konfigurierung in einer verteilten Unternehmensanwendung gleichzeitig Hochverfügbarkeit und Skalierbarkeit gewährleisten.

Aufgrund der für eine Sun Cluster-Umgebung erforderlichen Redundanz sorgt die Aufnahme von Sun Cluster in einer Lösung für eine merkliche Erhöhung der in der physischen Umgebung benötigten Anzahl von Computern und Netzwerkverbindungen.

Anders als die von anderen Java ES-Komponenten bereitgestellten Dienste, handelt es sich bei den Verfügbarkeitsdiensten von Sun Cluster um verteilte Peer-to-Peer-Dienste. Daher muss die Sun Cluster-Software auf jedem Computer in einem Cluster installiert werden.

Eine Erweiterung auf Sun Cluster-Software wird durch Sun Cluster Geographic Edition bereitgestellt, durch die Anwendungen mithilfe von mehreren geografisch voneinander getrennten Clustern und einer Infrastruktur, die Daten zwischen den Clustern repliziert, vor unerwarteten Störungen geschützt werden.

Hinweis – Sun Cluster und Sun Cluster Geographic Edition werden nur auf Solaris™-Betriebssystemen (Solaris OS) unterstützt.

Synthese der drei Architekturdimensionen

Zusammengenommen bilden die drei in Abbildung 2–1 dargestellten und in den letzten Abschnitten besprochenen Architekturdimensionen ein Framework für die Konzeption verteilter Softwarelösungen. Die drei Dimensionen (Infrastrukturdienstabhängigkeiten, logische Schichten und Dienstqualität) verdeutlichen die von den Java ES-Komponenten in Lösungsarchitekturen gespielte Rolle.

Jede Dimension steht für eine unterschiedliche Architekturperspektive. Eine Lösungsarchitektur muss alle Dimensionen berücksichtigen. Beispiel: Verteilte Komponenten in jeder logischen Schicht einer Lösungsarchitektur (Dimension 2) müssen durch entsprechende Infrastrukturkomponenten (Dimension 1) und entsprechende Dienstqualitätskomponenten (Dimension 3) unterstützt werden.

Entsprechend spielt jede in einer Lösungsarchitektur vorhandene Komponente hinsichtlich der verschiedenen Architekturdimensionen unterschiedliche Rollen. Directory Server kann beispielsweise sowohl als Back-End-Komponente der Datenschicht (Dimension 2) als auch als Anbieter von Persistenzdiensten (Dimension 1) angesehen werden. Aufgrund der Zentralität von Directory Server hinsichtlich dieser beiden Dimensionen stehen Dienstqualitätsprobleme (Dimension 3) für diese Java ES-Komponente an höchster Stelle. Ein Directory Server-Ausfall hätte schwerwiegende Auswirkungen auf ein Geschäftssystem, sodass ein Hochverfügbarkeitskonzept für diese Komponente von großer Wichtigkeit ist. Da Directory Server zur Speicherung geheimer Benutzer- oder Konfigurationsinformationen verwendet wird, ist ein Sicherheitskonzept für diese Komponente ebenfalls sehr wichtig.

Das Zusammenspiel der drei Dimensionen hinsichtlich der Java ES-Komponenten wirkt sich auf die Konzeption der logischen Architektur und der Bereitstellungsarchitektur der Lösungen aus.

Detaillierte Entwurfsmethodiken, die auf diesem durch „Java ES-Architektur-Framework" auf Seite 33 dargestellen Architektur-Framework basieren, werden in diesem Handbuch nicht beschrieben. Das dreidimensionale Architekturframework hebt jedoch Aspekte der Konzeption hervor, die für das Verständnis des Bereitstellens von auf Java Enterprise System basierenden Softwarelösungen wichtig sind.

Java ES-Lösungsarchitektur – Beispiel

Java ES unterstützt eine Vielzahl von Softwarelösungen. Viele Lösungen können mit den in Java ES enthaltenen Komponenten ohne Entwicklungsaufwand direkt konzipiert und bereitgestellt werden. Bei anderen Lösungen können umfangreiche Entwicklungsarbeiten notwendig sein, bei denen Sie eigene J2EE-Komponenten entwickeln müssen, die neue Geschäfts- oder Präsentationsdienste bereitstellen. Sie können diese benutzerdefinierten Komponenten in Form von Webdiensten zusammenfassen, die den SOAP-Schnittstellenstandards entsprechen. Bei vielen Lösungen ist eine Kombination dieser beiden Ansätze erforderlich.

In diesem Abschnitt finden Sie ein aus den im letzten Abschnitt beschriebenen Architekturkonzepten abgeleitetes Beispiel, das verdeutlicht, wie Java ES eine Out-of-the-Box-Lösung unterstützt.

Szenario der Unternehmenskommunikation

Unternehmen müssen in der Regel die Kommunikation zwischen Ihren Mitarbeitern, insbesondere durch E-Mail- und Kalenderdienste sicherstellen. Solche Unternehmen wollen, dass ihre Mitarbeiter einen personalisierten Zugang zu internen Websites und anderen Ressourcen besitzen, der auf den unternehmensweiten Authentifizierungs- und Autorisierungsdiensten basiert. Zusätzlich wollen diese Unternehmen, dass die Identität der Mitarbeiter über alle Unternehmensdienste hinweg protokolliert wird, damit eine einzige Webanmeldung (Single Sign-On) den Zugriff auf alle Dienste bietet.

Diese spezifischen Geschäftsanforderungen, die lediglich eine Beispielmenge der gesamten Geschäftsanforderungen darstellen, sind in der folgenden Tabelle zusammengefasst.

TABELLE 2–4 Zusammenfassung der Geschäftsanforderungen: Kommunikationsszenario

Geschäftsanforderung	Beschreibung	Erforderliche Dienste
Single Sign-On	Zugang zu sicheren Unternehmensressourcen und -diensten auf der Grundlage einer einzigen Identität mit Single Sign-On für Webzugang.	Identitätsdienste
Messaging Calendar	E-Mail-Messaging zwischen Mitarbeitern und mit der Außenwelt. Elektronische Kalendereinträge und Terminvereinbarungen für Mitarbeiter.	Kommunikations- und Zusammenarbeitsdienste:
Portalzugriff	Einzelner, webbasierter, personalisierter Zugangspunkt für Kommunikationsdienste, wie E-Mail und Kalender sowie interne Webseiten.	Zugangsdienste

Außerdem hat ein Unternehmen darüber hinaus Anforderungen hinsichtlich der Leistung, Verfügbarkeit, Netzwerksicherheit und Skalierbarkeit des Softwaresystems, das diese Dienste bereitstellt.

Logische Architektur des Beispielszenarios

Eine logische Architektur, mit der die in Tabelle 2–4 identifizierten Portal-, Kommunikations- und Identitätsdienste mithilfe von Java ES-Komponenten und Komponenten der Sun Java Communications Suite (Messaging Server, Calendar Server, Instant Messaging usw.) bereitgestellt werden, wird in folgender Abbildung dargestellt. Die Architektur behandelt logisch getrennte Konfigurationen von Messaging Server als separate Komponenten, da diese unterschiedliche Dienste bereitstellen.

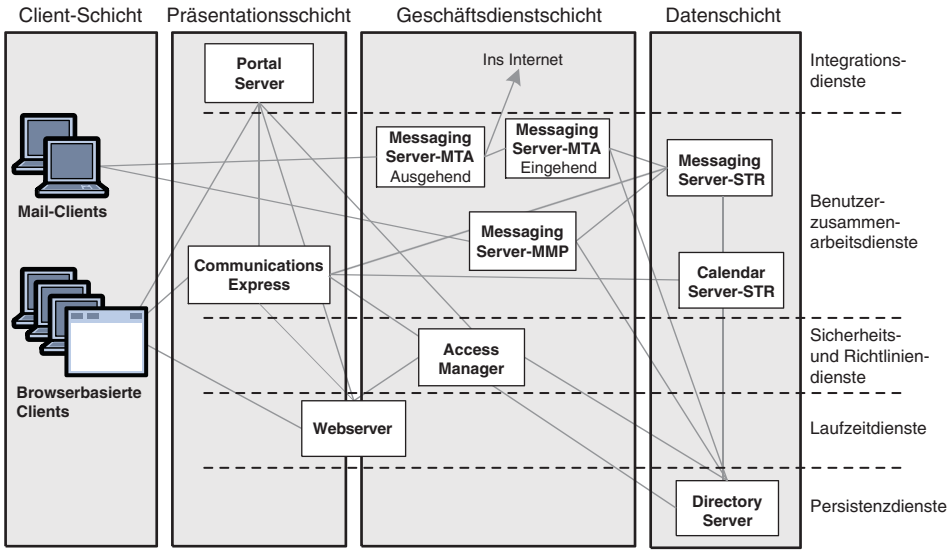

ABBILDUNG 2–6 Logische Architektur des Unternehmenskommunikationsszenarios

Die Komponenten sind in einer horizontalen Dimension platziert, die die standardmäßigen logischen Schichten darstellen. Die vertikale Dimension stellt die Infrastrukturdienstebenen dar. Die Interaktionen zwischen den Komponenten hängen entweder von ihrer Funktion als verteilte Infrastrukturdienste ab (Interaktionen zwischen Infrastrukturdienstebenen) oder von ihrer Rolle in einer Schichtenarchitektur der Anwendung (Interaktionen innerhalb und zwischen logischen Schichten).

In dieser Architektur arbeitet Access Manager, der auf die in Directory Server gespeicherten Benutzerinformationen zugreift, als Schiedsrichter hinsichtlich der Authentifizierung und Autorisierung über Single Sign-On für den Portal Server und andere webbasierte Komponenten der Präsentationsschicht. Zu den Messaging Server -Komponenten gehört in der Datenschicht

ein Nachrichtenspeicher (Messaging Server -STR), der Komponenten der Geschäftsdienstschicht sendet und abruft, sowie in der Präsentationsschicht eine HTTP-Zugangskomponente und Communications Express.

Die logische Architektur zeigt außerdem die Infrastrukturdienstabhängigkeiten zwischen den verschiedenen Komponenten. So hängt beispielsweise Portal Server für die Messaging- und Kalenderkanäle von Communications Express ab und für die Authentifizierungs- und Autorisierungsdienste von Access Manager. Diese Komponenten hängen wiederum von Directory Server ab, von dem Sie Benutzerinformationen und Konfigurationsdaten erhalten. Einige Komponenten benötigen Webcontainerdienste, die Web Server bereitstellt.

Weitere Informationen über das logische Konzept der Java ES-Lösung finden Sie im *Sun Java Enterprise System Deployment Planning Guide*.

Bereitstellungsarchitektur des Beispielszenarios

Beim Übergang von der logischen Architektur zur Bereitstellungsarchitektur sind die Dienstqualitätsanforderungen von größter Bedeutung. So können beispielsweise geschützte Subnetze und Firewalls genutzt werden, um eine Sicherheitsbarriere zum Back-End-Bereich zu erzeugen. Für viele Komponenten müssen Verfügbarkeits- und Skalierbarkeisanforderungen erfüllt werden, indem diese auf mehreren Computern bereitgestellt und die Anfragen über ein Lastausgleichsmodul an die replizierten Komponenten verteilt werden.

Wenn jedoch strengere Verfügbarkeitsanforderungen vorliegen und wenn sehr viel Festplattenspeicher benutzt wird, dann sind andere Verfügbarkeitslösungen besser geeignet. Für den Messaging Server -Speicher kann beispielsweise Sun Cluster und für Directory Server die Multi-Master-Replikation genutzt werden.

Weitere Informationen über das Bereitstellungskonzept der Java ES-Lösung finden Sie im *Sun Java Enterprise System Deployment Planning Guide*.

In diesem Kapitel enthaltene Schlüsselbegriffe

In diesem Abschnitt werden die wichtigsten in diesem Kapitel verwendeten technischen Bedingungen erläutert, wobei auf die Verwendung dieser Bedingungen im Zusammenhang mit Java ES besonders eingegangen wird.

Anwendungskomponente Eine kundenspezifisch entwickelte Software-Komponente, die eine spezifische Datenverarbeitungsfunktionen durchführt und Endbenutzer oder andere Anwendungskomponenten mit Geschäftsdiensten versorgt. Eine Anwendungskomponente entspricht normalerweise einem verteilten Komponentenmodell (z. B. CORBA und der J2EE-Plattform). Diese Komponenten können (einzeln oder kombiniert) als web services zusammengefasst werden.

Architektur Ein Konzept, das die logischen und physischen modularen Blöcke einer verteilten Anwendung (oder eines anderen Softwaresystems) sowie ihre Beziehungen untereinander darstellt. Für eine Verteilte Unternehmensanwendung umfasst das Architekturkonzept im Allgemeinen sowohl die Logische Architektur als auch die Bereitstellungs- architektur der Anwendung.

Geschäftsdienst Eine Anwendungskomponente oder Komponentengruppe, die die Geschäftslogik im Namen mehrerer Clients ausführt (und daher einen Vorgang mit mehreren Threads darstellt). Ein Geschäftsdienst kann auch eine Gruppe von verteilten Komponenten sein, die zu einem Webdienst oder zu einem eigenständigen Server zusammengefasst sind.

Client Eine Software, die Software dienste anfordert. Ein Client kann ein Dienst sein, der einen anderen Dienst anfordert, oder eine GUI-Komponente, auf die ein Endbenutzer zugreift.

Bereitstellungs-architektur Ein starkes Konzept, das die Logische Architektur einer physischen Computerumgebung zuordnet. Die physische Umgebung umfasst die Computer in einer Intranet- oder Internetumgebung, die Netzwerkverbindungen zwischen ihnen sowie andere physische Geräte, die zur Unterstützung der Software erforderlich sind.

Logische Architektur Ein Konzept, das die logischen modularen Blöcke einer verteilten Anwendung sowie ihre Beziehungen untereinander (bzw. ihre Schnittstellen) darstellt. Die logische Architektur umfasst sowohl die verteilten application components als auch die Infrastrukturdienste, die für deren Unterstützung erforderlich sind.

Server Ein Softwarevorgang mit mehreren Threads (im Gegensatz zu einem Hardwareserver), der einen verteilten service oder eine geschlossene Gruppe von Diensten für clients bereitstellt, die über eine externe Schnittstelle auf den Dienst zugreifen

Webdienst Ein Dienst, der den standardisierten Internetprotokollen für Verfügbarkeit, Dienstintegrierung und Erkennung entspricht. Zu diesen Standards gehören das SOAP-Nachrichtenprotokoll, die WSDL-Schnittstellendefinition (Web Service Definition Language) und der UDDI-Registrierungsstandard (Universal Discovery, Description and Integration).

◆ ◆ ◆ KAPITEL 3

Java ES-Integrationsfunktionen

Dieses Kapitel beschreibt den konzeptionellen und technischen Hintergrund der Funktionen, die bei der Integration von Java ES-Komponenten zu einem einzigen Softwaresystem eine Schlüsselrolle spielen. Anhand dieser Funktionen werden die Vorteile von Java ES im Vergleich zur manuellen Integration unvereinbarer Infrastrukturprodukte deutlich.

Das Kapitel enthält die folgenden Abschnitte:

Das integrierte Installationsprogramm von Java ES

Alle Java ES-Komponenten werden mithilfe eines einzigen Installationsprogramms installiert. Das Java ES-Installationsprogramm überträgt Java ES-Software an ein Hostsystem. Mit dem Installationsprogramm können Sie bestimmte Java ES-Komponenten auswählen und auf einem Hostcomputer in Ihrer Datenverarbeitungsumgebung installieren. Das Installationsprogramm bietet auch einige Möglichkeiten zur Konfiguration des Installationszeitpunkts, die von den einzelnen zu installierenden Java ES-Komponenten abhängen.

Das Java ES-Installationsprogramm führt selbstständig keine verteilten Installationen durch. Um eine verteilte Java ES-Softwarelösung bereitzustellen, installieren Sie die entsprechenden Komponenten mit dem Java ES-Installationsprogramm nacheinander auf jedem Computer Ihrer Umgebung. Abhängig von Ihrer Bereitstellungsarchitektur und den Abhängigkeiten zwischen den Komponenten müssen Sie eine Reihe von Installationssitzungen und Konfigurationsschritten durchführen.

Das Installationsprogramm arbeitet interaktiv sowohl in einem Grafikmodus als auch in einem textbasierten Modus und verfügt außerdem über einen parametergesteuerten Modus zur

automatischen Installation. Neben Englisch unterstützt das Installationsprogramm die folgenden Sprachen: Französisch, Deutsch, Japanisch, Koreanisch, Spanisch, Chinesisch (vereinfacht) und Chinesisch (traditionell).

In diesem Abschnitt werden Aspekte des Java ES-Installationsprogramms behandelt. Detailliertere Informationen erhalten Sie im *Sun Java Enterprise System 5 Installationshandbuch für UNIX*.

Überprüfung bereits vorhandener Software

Das Installationsprogramm überprüft den Hostcomputer, auf dem die Installation durchgeführt wird, und ermittelt die bereits installierten Java ES-Komponenten. Das Installationsprogramm führt dann Prüfungen auf verschiedenen Ebenen durch, um sicherzustellen, dass alle vorhandenen Komponenten die für eine erfolgreiche Zusammenarbeit erforderliche Versionsstufe aufweisen. Das Installationsprogramm informiert Sie darüber, welche Softwarekomponenten nicht kompatibel sind und aufgerüstet oder entfernt werden müssen.

Ebenso überprüft das Installationsprogramm, ob bereits gemeinsam genutzte Java ES-Komponenten installiert wurden, beispielsweise J2SE oder NSS, und führt sämtliche Inkompabilitäten auf (siehe „Gemeinsam genutzte Komponenten" auf Seite 23). Wenn Sie mit der Installation fortfahren, rüstet das Installationsprogramm die gemeinsam genutzten Komponenten auf eine neuere Version auf.

Abhängigkeitsüberprüfung

Das Installationsprogramm führt eine intensive Überprüfung der Komponenten durch, um festzustellen, ob die von Ihnen ausgewählten Installationskomponenten ordnungsgemäß funktionieren. Etliche Komponenten sind von anderen Komponenten abhängig. Aus diesem Grund schließt das Installationsprogramm, wenn Sie eine Komponente für die Installation auswählen, automatisch die Komponenten und Unterkomponenten mit ein, zu denen die ausgewählte Komponente in einem Abhängigkeitsverhältnis steht. Sie können die Auswahl einer Komponente nicht aufheben, wenn eine andere ausgewählte Komponente lokal von dieser Komponente abhängt. Wenn die Abhängigkeit jedoch nicht auf lokaler Ebene besteht, wird eine Warnmeldung ausgegeben. Sie können den Vorgang jedoch fortsetzen (es wird davon ausgegangen, dass die Abhängigkeitsanforderung durch eine Komponente auf einem anderen Hostcomputer erfüllt wird).

Erstkonfiguration

Bei vielen Java ES-Komponenten ist eine Erstkonfiguration erforderlich, damit diese gestartet werden können. Bei einigen Komponenten wird das Java ES-Installationsprogramm verwendet, um diese Erstkonfiguration vorzunehmen.

Sie können auswählen, ob das Installationsprogramm die Erstkonfiguration durchführen (Option "Jetzt konfigurieren") oder die Software ohne Erstkonfiguration installieren soll (Option "Später konfigurieren"). Im letzten Fall müssen Sie jede Komponente nach Abschluss der Installation explizit konfigurieren.

Wenn Sie auswählen, dass das Installationsprogramm die Erstkonfiguration vornehmen soll, müssen Sie während des Installationsvorgangs die benötigten Konfigurationsangaben vornehmen. Insbesondere können Sie eine Reihe von Parameterwerten angeben, die für alle Produktkomponenten gelten sollen, beispielsweise eine Administrator-ID und das zugehörige Passwort.

Deinstallation

Java ES stellt darüber hinaus ein Deinstallationsprogramm bereit, mit dem Komponenten entfernt werden können, die vom Java ES-Installationsprogramm auf dem lokalen Computer installiert wurden. Das Deinstallationsprogramm prüft, ob lokale Abhängigkeiten vorliegen, und gibt gegebenenfalls eine Warnmeldung aus. Das Deinstallationsprogramm entfernt keine gemeinsam genutzten Java ES-Komponenten. Ebenso wie das Installationsprogramm kann das Deinstallationsprogramm im grafischen, textbasierten oder Automatikmodus ausgeführt werden.

Systemüberwachungsdienste

Java ES beinhaltet eine neue Überwachungsfunktion, die eine Echtzeitüberwachung von Systemdiensten bereitstellt. Die Überwachung wird durch Sun Java System Monitoring Framework (Gemeinsam genutzte Komponente) und die Sun Java System Monitoring Console (Produktkomponente) implementiert Das Monitoring Framework wird automatisch konfiguriert und aktiviert, um Daten für jede installierte Java ES-Komponente zusammenzustellen und die Monitoring Console stellt die grafische Schnittstelle dar, die zur Anzeige der überwachten Daten verwendet wird. Die Monitoring Console ist eine Komponente, die während der Installation von Java ES aktiviert werden kann, und das Monitoring-Framework wird automatisch installiert.

Unter der Überwachung versteht man den Prozess für das Zusammenstellen und Veröffentlichen von Laufzeitdaten und zur Berechnung von Qualitätsdienstkriterien, sodass die Systemadministratoren die Leistung überprüfen und Alarme empfangen können. Während Laufzeitvorgängen interagieren die Administratoren mit der Monitoring Console, um Leistungsstatistiken anzuzeigen, Schwellenwerte für die dynamische Überwachung festzulegen, benutzerdefinierte Überwachungsaufgaben zu definieren und Alarme zu bestätigen.

Integrierte Identitäts- und Sicherheitsdienste

Eine wichtige Funktion von Java ES besteht in der integrierten Verwaltung von Benutzeridentitäten und dem integrierten Authentifizierungs- und Autorisierungs-Framework. In folgendem Abschnitt finden Sie den technischen Hintergrund für das Verständnis der integrierten Identitäts- und Sicherheitsdienste, die Java ES bietet:

Einzelidentität

Innerhalb einer Java ES-Umgebung verfügt jeder Benutzer über eine einzelne integrierte Identität. Anhand dieser Einzelidentität kann dem Benutzer der Zugriff auf verschiedene Ressourcen gestattet werden, beispielsweise Portale, Webseiten und Dienste wie Nachrichtendienste, Kalenderdienste und Instant Messaging.

Diese integrierte Identitäts- und Sicherheitsfunktion beruht auf einer engen Zusammenarbeit zwischen Directory Server, Access Manager und anderen Java ES-Komponenten.

Der Benutzerzugriff auf einen Dienst oder eine Ressource von Java ES wird gewährt, indem die benutzerspezifischen Informationen in einem einzelnen Benutzereintrag in einem Benutzer-Repository oder Verzeichnis gespeichert werden. Zu diesen Informationen gehören für gewöhnlich Daten wie ein eindeutiger Name und ein Passwort, eine E-Mail-Adresse, die Funktion innerhalb einer Organisation, Webseiteneinstellungen usw. Die Informationen im Benutzereintrag können zur Authentifizierung des Benutzers, zur Autorisierung des Zugriffs auf bestimmte Ressourcen und zur Bereitstellung verschiedener Dienste für den jeweiligen Benutzer verwendet werden.

Bei Java ES werden die Benutzereinträge in einem von Directory Server bereitgestellten Verzeichnis gespeichert. Wenn ein Benutzer einen von einer Java ES-Komponente bereitgestellten Dienst anfordern möchte, verwendet dieser Dienst Access Manager für die Authentifizierung des Benutzers und für die Autorisierung des Zugriffs auf bestimmte Ressourcen. Der angeforderte Dienst prüft die im Verzeichniseintrag des Benutzers enthaltenen benutzerspezifischen Konfigurationsangaben. Der Dienst verwendet diese Informationen, um die vom Benutzer angeforderten Aufgaben durchzuführen.

Die folgende Abbildung illustriert den Zugriff auf Benutzereinträge für die Durchführung der Benutzerauthentifizierung und -autorisierung hinsichtlich der Bereitstellung eines Dienstes für den Benutzer.

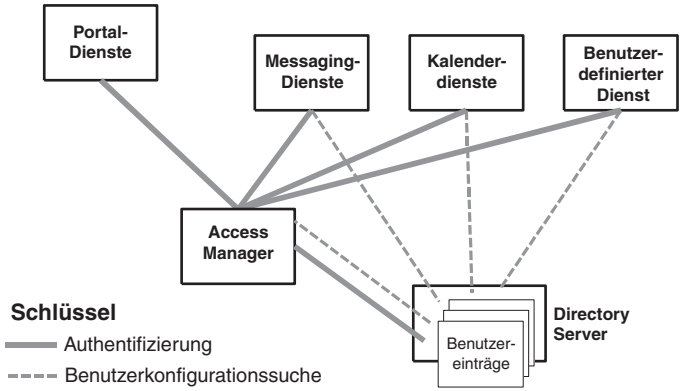

ABBILDUNG 3–1 Ein einzelner Benutzereintrag unterstützt viele Dienste

Eine der sich aus diesem System ergebenden Funktionen besteht darin, dass sich ein Benutzer im Web bei jedem Java ES-Dienst anmelden kann und dadurch automatisch bei den anderen Diensten des Systems authentifiziert wird. Diese leistungsstarke Java ES-Funktion ist unter der Bezeichnung Single Sign-On bekannt.

Authentifizierung und Single Sign-On

Access Manager stellt Java ES-Authentifizierungs- und Autorisierungsdienste bereit. Access Manager verwendet Informationen aus Directory Server, um als Broker für die Interaktion von Benutzern mit Java ES-Webdiensten und anderen webbasierten Diensten in einem Unternehmen zu fungieren.

Access Manager verwendet außerdem eine externe Komponente, den so genannten Richtlinienagenten. Der Richtlinienagent stellt eine Verbindung zu dem Webserver her, der als Host für einen Dienst bzw. eine Ressource fungiert, die durch Access Manager gesichert ist. Der Richtlinienagent schaltet sich bei Anforderungen des Benutzers an die gesicherten Ressourcen im Namen von Access Manager ein. Bei einigen Java ES-Komponenten wie beispielsweise dem Portal Server wird der Richtlinienagent von der Unterkomponente Access Manager SDK bereitgestellt.

Authentifizierung

Access Manager beinhaltet einen Authentifizierungsdienst zur Überprüfung der Identitäten der Benutzer, die (über HTTP oder HTTPS) Zugriff auf Webdienste innerhalb eines Unternehmens anfordern. Beispiel: Ein Firmenangestellter, der die Telefonnummer eines Kollegen nachschlagen muss, verwendet einen Browser, um zum Online-Telefonbuch des Unternehmens zu gelangen. Um sich beim Telefonbuchdienst anzumelden, muss der Benutzer eine Benutzer-ID und ein Passwort angeben.

Die Authentifzierungssequenz ist in Abbildung 3–2 dargestellt. Ein Richtlinienagent schaltet sich in die Anfrage, die eine Anmeldung beim Telefonbuch enthält, ein (1) und schickt eine Anfrage an den Authentifizierungsdienst (2). Der Authentifizierungsdienst prüft die Benutzer-ID und das Kennwort anhand der in Directory Server speicherten Informationen (3). Wenn die Anmeldeanforderung gültig ist, wird der Benutzer authentifiziert (4, 5 und 6) und dem Mitarbeiter wird das Firmentelefonbuch angezeigt (7). Wenn die Anmeldeanforderung nicht gültig ist, wird eine Fehlermeldung generiert und die Authentifizierung schlägt fehl.

Der Authentifizierungsdienst unterstützt auch zertifikatbasierte Authentifizierungen über HTTPS.

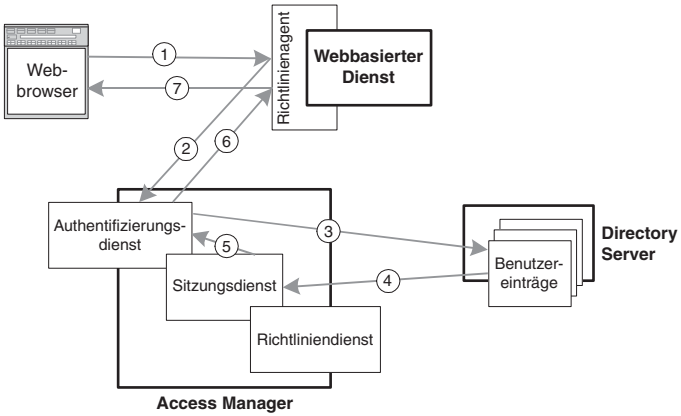

ABBILDUNG 3–2 Authentifizierungssequenz

Single Sign-On

In dem in den vorherigen Abschnitten beschriebenen Authentifizierungsszenario fehlt ein wichtiger Schritt. Wenn die Authentifizierungsanfrage eines Benutzers überprüft ist, wird der Sitzungsdienst von Access Manager aufgerufen (4), wie in Abbildung 3–2 dargestellt. Der Sitzungsdienst erstellt ein Sitzungs-Token, das Identitätsinformationen des Benutzers und eine Token-ID enthält (5). Das Sitzungs-Token wird an den Richtlinienagenten zurückgesendet (6), der es (als Cookie) an den Browser weiterleitet (7), von dem die Authentifizierungsanforderung ausging.

Wenn der authentifizierte Benutzer versucht, auf einen anderen gesicherten Dienst zuzugreifen, leitet der Browser das Sitzungs-Token an den entsprechenden Richtlinienagenten weiter. Der Richtlinienagent überprüft mit dem Sitzungsdienst, ob die frühere Authentifizierung des Benutzers noch gültig ist, und der Benutzer erhält Zugriff auf den zweiten Dienst, ohne dass er noch einmal eine Benutzer-ID und ein Passwort eingeben muss.

Ein Benutzer muss sich daher nur ein einziges Mal anmelden, um bei mehreren von Java ES bereitgestellten webbasierten Diensten authentifiziert zu werden. Die Single Sign-On-Authentifizierung bleibt gültig, bis sich der Benutzer explizit abmeldet oder die Sitzung abläuft.

Autorisierung

Access Manager beinhaltet auch einen Richtliniendienst, mit dem der Zugriff auf webbasierte Ressourcen in einer Java ES-Umgebung gesteuert werden kann. Eine Richtlinie ist eine Regel, die angibt, wer autorisiert ist, unter bestimmten Bedingungen auf eine bestimmte Ressource zuzugreifen. Die Autorisierungssequenz ist in der folgenden Abbildung dargestellt.

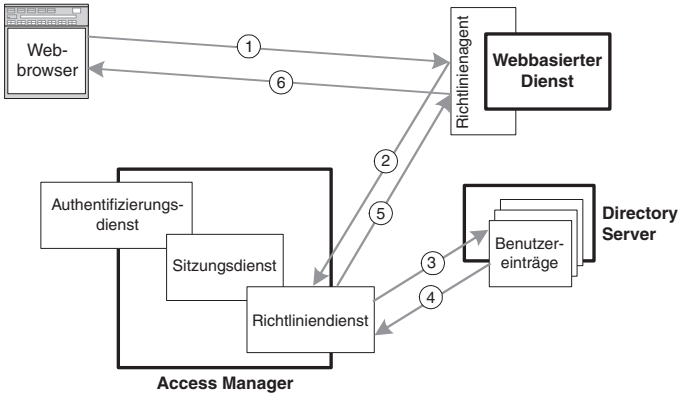

ABBILDUNG 3–3 Autorisierungssequenz

Wenn ein authentifizierter Benutzer eine Anforderung für eine durch Access Manager gesicherte Ressource abschickt (1), benachrichtigt der Richtlinienagent den Richtliniendienst (2), der anhand der in Directory Server vorhandenen Informationen (3) die Zugriffsrichtlinie für die Ressource prüft, um festzustellen, ob der Benutzer über die erforderlichen Zugriffsberechtigungen verfügt (4). Wenn der Benutzer zugriffsberechtigt ist (5), wird die Ressourcenanfrage ausgeführt (6).

Access Manager bietet die Mittel, die erforderlich sind, um innerhalb eines Unternehmens Richtlinien zu definieren, zu bearbeiten, zu gewähren, zu widerrufen und zu löschen. Die Richtlinien werden in Directory Server gespeichert und über richtlinienbezogene Attribute in Organisationseinträgen konfiguriert. Es können auch Rollen für Benutzer definiert und in Richtliniendefinitionen integriert werden.

Die Access Manager-Richtlinienagenten sind für die Durchsetzung der Richtlinien zuständig. Wenn der Richtliniendienst eine Zugriffsanforderung zurückweist, verweigert der Richtlinienagent dem betreffenden Benutzer den Zugriff auf die gesicherten Ressourcen.

In diesem Kapitel enthaltene Schlüsselbegriffe

In diesem Abschnitt werden die wichtigsten in diesem Kapitel verwendeten technischen Bedingungen erläutert, wobei auf die Verwendung dieser Bedingungen im Zusammenhang mit Java ES besonders eingegangen wird.

Verzeichnis Eine spezielle Art von Datenbank, die für das Lesen (und weniger das Schreiben) von Daten optimiert ist. Die meisten Verzeichnisse beruhen auf LDAP (Lightweight Directory Access Protocol), einem als Industriestandard etablierten Protokoll.

Richtlinie Eine Richtlinie ist eine Regel, die angibt, wer autorisiert ist, unter bestimmten Bedingungen auf eine bestimmte Ressource zuzugreifen. Diese Regel kann auf Gruppen von Benutzern oder Rollen in einer Organisation basieren.

Einzelidentität Eine Identität, über die ein Benutzer dank eines einzelnen Benutzereintrags in einem Java ES-Verzeichnis verfügt. Anhand dieser einzelnen Benutzeridentität kann einem Benutzer der Zugriff auf verschiedene Java ES-Ressourcen gestattet werden, beispielsweise auf Portale, Webseiten und Dienste, wie Nachrichtendienste, Kalenderdienste und Instant Messaging.

Single Sign-On Eine Funktion, die es ermöglicht, die Authentifizierung eines Benutzers bei einem Dienst in einem verteilten System automatisch auf andere Dienste in diesem System anzuwenden.

◆ ◆ ◆ KAPITEL 4

Java ES-Lösungslebenszyklus

In diesem Kapitel werden die Konzepte und die dazugehörige Terminologie besprochen, die für die einzelnen Phasen des Java ES-Lösungslebenszyklus relevant sind. Der Schwerpunkt liegt dabei auf den Bereitstellungsaufgaben, insbesondere auf dem Bereitstellungskonzept und den Aufgaben der Bereitstellungsimplementierung.

In diesem Kapitel werden die Aufgaben beschrieben, die in den einzelnen Phasen des Lebenszyklus anfallen. Das Kapitel enthält die folgenden Abschnitte:

Aufgaben des Lösungslebenszyklus

Der Lösungslebenszyklus wurde in Kapitel 1 als ein Standardansatz für die Implementierung von Geschäftslösungen mit der Java ES-Software vorgestellt. Das Lebenszyklusdiagramm wird zu Referenzzwecken noch einmal wiederholt.

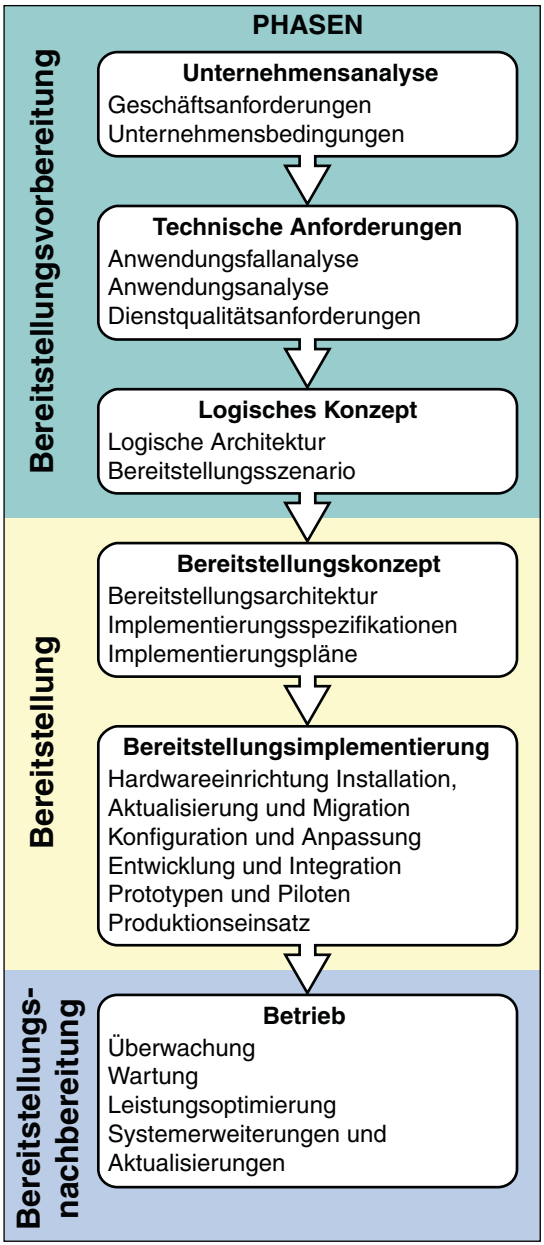

ABBILDUNG 4–1 Aufgaben des Lösungslebenszyklus

Bereitstellungsvorbereitung

In den Phasen der Bereitstellungs- vorbereitung des Lebenszyklus übertragen Sie die Analyse der Geschäftsanforderungen in ein Bereitstellungs- szenario. Das Bereitstellungsszenario dient als Spezifikation für ein Bereitstellungskonzept.

Die Aufgaben der Bereitstellungsvorbereitung werden in drei Phasen aufgeteilt, wie in Abbildung 4–1 gezeigt:

- **Unternehmensanalyse:** Definieren Sie die Geschäftsziele eines Bereitstellungsvorschlags und nennen Sie die Geschäftsanforderungen und -beschränkungen, die zur Erreichung dieses Ziels erfüllt werden müssen.
- **Technische Anforderungen:** Verwenden Sie die Ergebnisse der Geschäftsanalyse, um Anwendungsfälle zu erstellen, die die Interaktion der Benutzer mit einem vorweggenommenen Softwaresystem abbilden. Außerdem legen Sie die für diese Anwendungsfälle erwarteten Anwendungsmuster fest. Unter Berücksichtigung der Geschäftsanalyse und der Anwendungsanalyse formulieren Sie Dienstqualitätsanforderungen (siehe Tabelle 2–2), die der Bereitstellungsvorschlag einhalten muss.
- **Logisches Konzept:** Analysieren Sie die in der Phase der technischen Anforderungen entwickelten Anwendungsfälle, um festzustellen, welche Java ES-Infrastrukturkomponenten und welche kundenspezifisch entwickelten Anwendungskomponenten Sie für die Endbenutzer benötigen. Anhand der in Kapitel 2 besprochenen Konzepte entwerfen Sie eine logische Architektur. Die logische Architektur enthält alle Komponenten und alle zwischen Komponenten stattfindenden Interaktionen, die notwendig sind, damit die Anwendungsfälle einer bestimmten Softwarelösung umgesetzt werden.

Die logische Architektur wird zusammen mit Dienstqualitätsanforderungen wie Leistung, Verfügbarkeit und Sicherheit in einem Bereistellungsszenario zusammengefasst, wie in der folgenden Abbildung gezeigt. Weitere Informationen zu den Phasen der Bereitstellungsvorbereitung des Lebenszyklus finden Sie im *Sun Java Enterprise System Deployment Planning Guide*.

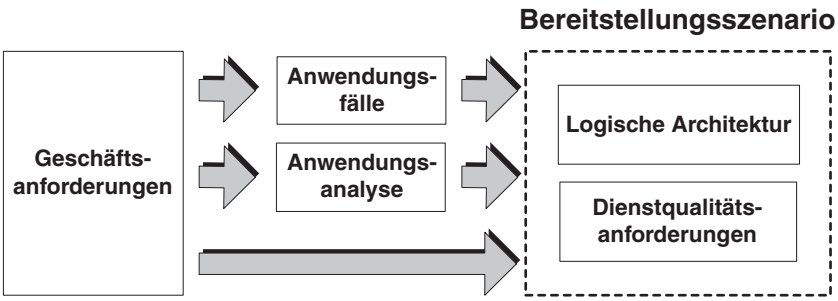

ABBILDUNG 4–2 Spezifizieren eines Bereitstellungsszenarios

Bereitstellung

In den Phasen Bereitstellung des Lebenszyklus übertragen Sie ein Bereitstellungsszenario in ein Bereitstellungskonzept, das anschließend implementiert, getestet und in einer Produktionsumgebung eingesetzt wird.

Der Bereitstellungsprozess umfasst im Allgemeinen Softwarekomponenten aller Schichten und aller Infrastrukturdienstebenen, die zur Unterstützung einer Softwarelösung erforderlich sind. Grundsätzlich müssen Sie kundenspezfisch entwickelte Anwendungskomponenten (J2EE-Komponenten, Webdienste oder andere Server) und die für die Lösung benötigten Java ES-Komponenten bereitstellen.

Die Aufgaben der Bereitstellung werden in zwei Phasen aufgeteilt, wie in Abbildung 4–1 gezeigt:

- „Bereitstellungskonzept" auf Seite 64. Das Bereitstellungskonzept hängt sowohl von der logischen Architektur einer Lösung als auch von der Leistung, Verfügbarkeit, Sicherheit, Skalierbarkeit, Zweckmäßigkeit und anderen Dienstqualitätsanforderungen ab, die eine Lösung erfüllen muss. Die Dimension der Dienstqualität einer Bereitstellungsarchitektur spielt in der Phase des Bereitstellungskonzepts eine wichtige Rolle.
- „Bereitstellungsimplementierung" auf Seite 67. Die Implementierung eines Bereitstellungskonzepts stellt einen iterativ verlaufenden Prozess dar, der die Einrichtung der Hardware, die Installation und Konfiguration der Software, die Entwicklung und Integration, das Testen und andere Aspekte des Produktionseinsatzes umfasst.

  Diese beiden Phasen des Bereitstellungsprozesses werden in den folgenden Abschnitten untersucht.

Bereitstellungskonzept

In der Bereitstellungskonzeptphase erstellen Sie eine Bereitstellungsarchitektur auf hoher Ebene, gefolgt von Implementierungsspezifikationen auf niedriger Ebene.

Bereitstellungsarchitektur

Eine Bereitstellungsarchitektur wird durch Zuordnung der logischen Modulblöcke einer Anwendung (die logische Architektur) zu einer physischen Computerumgebung erstellt. Hierbei müssen die im Bereitstellungsszenario festgelegten Dienstqualitätsanforderungen erfüllt sein. Das Bereitstellungsszenario wird, wie in der folgenden Abbildung dargestellt, in eine Bereitstellungsarchitektur umgesetzt.

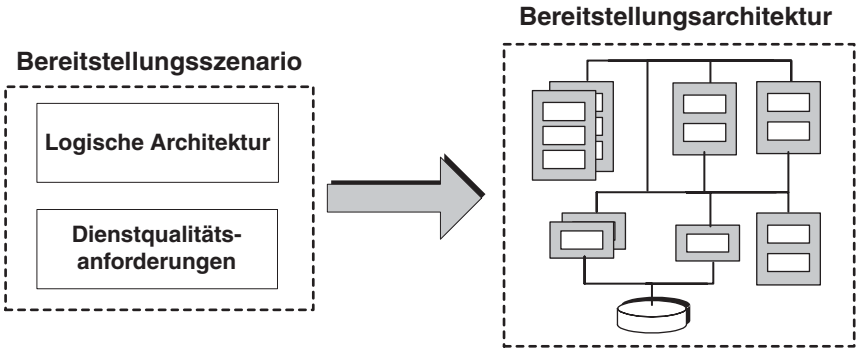

ABBILDUNG 4–3 Umsetzen eines Bereitstellungsszenarios in eine Bereitstellungsarchitektur

Ein Aspekt dieses Architekturkonzepts bildet die Festlegung der Größe der physischen Umgebung (Bestimmung der Anzahl der Computer und Schätzung ihrer Prozessorstärke und RAM-Anforderungen), die notwendig ist, um die Anforderungen hinsichtlich der Leistung, Verfügbarkeit, Sicherheit und anderer Dienstqualitätsanforderungen zu erfüllen. Nachdem Sie die Größe festgelegt haben, weisen Sie den Computern in der physischen Umgebung Java ES-Komponenten und Anwendungskomponenten zu. Die resultierende Bereitstellungsarchitektur muss die Funktionen der verschiedenen Computer, die Eigenschaften der Systeminfrastrukturdienste und die Beschränkungen für die Betriebs- oder Verfügbarkeitskosten berücksichtigen.

Je größer die Anzahl der Java ES-Komponenten im Bereitstellungsszenario ist und je höher die Dienstqualitätsanforderungen sind, desto höher sind die Ansprüche, die an die Computer und die Netzwerkbandbreite gestellt werden. Wenn die Hardware begrenzt oder extrem teuer ist, müssen Sie eventuell Kompromisse zwischen den Festkosten (Hardware) und den variablen Kosten (Personalbedarf) oder zwischen den verschiedenen Dienstqualitätsanforderungen schließen. Möglicherweise müssen Sie aber auch Ihr Konzept vereinfachen.

Die Konzeption einer Bereitstellungsarchitektur erfolgt oft in sich wiederholenden Schritten. Referenzbereitstellungsarchitekturen dienen als Ausgangspunkt für Java ES-Bereitstellungskonzepte.

Eine Referenzarchitektur basiert auf einem bestimmten Bereitstellungsszenario: Eine logische Architektur mit bestimmten Dienstqualitätsanforderungen. In der Referenzarchitektur wird eine Softwarelösung so innerhalb einer bestimmten physischen Umgebung bereitgestellt, dass die vorgegebenen Dienstqualitätsanforderungen erfüllt werden. Das Testen der Leistung bei

vorgegebenen Lasten basiert auf derselben Gruppe von Anwendungsfällen, aus denen das Bereitstellungsszenario entwickelt wurde. Dokumentation zur Referenzarchitektur ist für Java ES-Kunden öffentlich verfügbar.

Auf der Grundlage einer Referenz-Bereitstellungsarchitektur oder einer Kombination von Referenzarchitekturen können Sie eine erste näherungsweise Bereitstellungsarchitektur entwerfen, die Ihre eigenen Bereitstellungsszenario-Anforderungen erfüllt. Sie können unter Berücksichtigung der Unterschiede zwischen Ihrem eigenen Bereitstellungsszenario und den Bereitstellungsszenarios, auf denen die Referenzarchitekturen basieren, die Referenzarchitekturen anpassen oder als Referenzpunkte verwenden. Auf diese Weise können Sie die Auswirkungen Ihrer eigenen Größen-, Leistungs-, Sicherheits-, Verfügbarkeits-, Kapazitäts- und Zweckmäßigkeitsanforderungen beurteilen.

Implementierungsspezifikationen

Implementierungsspezifikationen enthalten die für die Implementierung einer Bereitstellungsarchitektur notwendigen Einzelheiten. Die Spezifikationen enthalten in der Regel folgende Informationen:

- Vorhandene Hardware, mit Computern, Speichergeräten, Lastausgleichsmodulen und Netzwerkkabeln
- Betriebssysteme
- Netzwerkstruktur, mit Subnetzen und Sicherheitszonen
- Einzelheiten des Verfügbarkeitskonzepts
- Einzelheiten des Sicherheitskonzepts
- Für das Einrichten von Endbenutzern benötigte Informationen über das Verzeichniskonzept.

Implementierungspläne

Implementierungspläne beschreiben, wie Sie die Durchführung der verschiedenen Aufgaben der Phase der Bereitstellungsimplementierung geplant haben. Die Pläne enthalten in der Regel folgende Aufgaben:

- Einrichten der Hardware
- Installation, Aktualisierung und Migration der Software
- Konfiguration und Anpassung des Systems
- Entwicklung und Integration
- Testen
- Produktionseinsatz

Bereitstellungsimplementierung

Die Implementierung eines Bereitstellungskonzepts besteht aus den im vorherigen Abschnitt aufgeführten Aufgaben, die auch in Abbildung 4–1 dargestellt sind. Die Reihenfolge dieser Aufgaben ist nicht streng vorgegeben, da der Bereitstellungsprozess als solcher iterativ verläuft. In den nachfolgenden Unterabschnitten werden die wichtigsten Aufgaben der Bereitstellungsimplementierung in der Reihenfolge beschrieben, in der sie üblicherweise ausgeführt werden.

Einrichten der Hardware

Die Implementierungsspezifikationen enthalten alle Details Ihrer physischen Umgebung: Computer, Netzwerkstruktur, Netzwerkhardware (mit Leitungen, Schaltern, Routern und Lastausgleichsmodulen), Speichergeräte usw. Die gesamte Hardware muss als Plattform für Ihre Java ES-Lösung eingerichtet werden.

Installation, Aktualisierung und Migration der Software

Aus der Bereitstellungsarchitektur und den in den Implementierungsspezifikationen enthaltenen zusätzlichen Details erfahren Sie, welche Anwendungskomponenten und welche Java ES-Komponenten auf jedem Computer Ihrer physischen Umgebung vorhanden sein müssen. Mit dem in Java ES integrierten Installationsprogramm installieren Sie auf jedem Computer Ihrer Bereitstellungsarchitektur die entsprechenden Java ES-Komponenten (siehe „Das integrierte Installationsprogramm von Java ES" auf Seite 53).

Ihr Installationsplan beschreibt den Ablauf und den Umfang der Installationssitzungen. Nach welchem Ansatz Sie die Installationen durchführen, kann jedoch davon abhängen, ob Sie eine Neuinstallation von Java ES durchführen oder ob Sie zuvor bereits installierte Java ES-Komponenten aktualisieren oder ob Sie Komponenten anderer Hersteller durch Java ES ersetzen. Bei den beiden letztgenannten Java ES-Einführungsszenarios ist es häufig notwendig, dass aus Gründen der Kompatibilität Daten oder Anwendungscode migriert werden müssen.

Konfiguration und Anpassung des Systems

Damit die verschiedenen Systemkomponenten als integriertes System zusammenarbeiten, muss die Systemkonfiguration an einigen Stellen angepasst werden. Dazu gehört zuerst die für den Start jeder Systemkomponente notwendige Erstkonfiguration. Anschließend müssen alle Java ES-Komponenten so konfiguriert werden, dass sie mit den Komponenten kommunizieren, mit denen sie interagieren.

Entsprechend der für jede Komponente vorliegenden Verfügbarkeitslösung muss darüber hinaus die Hochverfügbarkeit konfiguriert werden. Die Benutzer müssen eingerichtet werden, damit sie auf die verschiedenen Dienste zugreifen können. Darüber hinaus müssen die Richtlinien und die Steuerung für die Authentifizierung und die Autorisierung eingerichtet werden (siehe „Integrierte Identitäts- und Sicherheitsdienste" auf Seite 56).

In den meisten Fällen gehören zu den Konfigurationsaufgaben auch gewisse Anpassungen der Java ES-Komponenten, damit exakt die benötigten Funktionen verfügbar sind. Sie passen beispielsweise Portal Server für gewöhnlich an, um Anschlusskanäle verfügbar zu machen und Access Manager, um Autorisierungsaufgaben durchzuführen usw.

Entwicklung und Integration

Die logische Architektur, die im Bereitstellungsszenario festgelegt ist, bestimmt das Ausmaß der kundenspezifischen Entwicklungarbeit, die für die Implementierung einer Lösung erforderlich ist.

Für einige Bereitstellungen kann die Entwicklungsarbeit viel umfangreicher sein. Möglicherweise müssen Sie unter Verwendung von J2EE-Komponenten, die in einer Anwendungsserver- oder Web Server-Umgebung ausgeführt werden, ganz neue Geschäfts- und Präsentationsdienste entwickeln. In diesen Fällen erstellen Sie einen Prototyp für die Lösung und testen das Konzept, bevor Sie mit der vollständigen Entwicklung anfangen.

Für Lösungen, bei denen umfangreiche Entwicklungsarbeiten notwendig sind, bietet Sun Java™ Studio verschiedene Tools für die Programmierung verteilter Komponenten oder Geschäftsdienste. Die Entwickler-Tools von Sun Java Studio vereinfachen die Programmierung und das Testen von Anwendungen, die von der Java ES-Infrastruktur unterstützt werden.

In bestimmten Situationen müssen Java ES-Komponenten mit Legacy-Anwendungen oder Diensten anderer Hersteller integriert werden. Eine solche Integration kann in der Datenschicht vorhandene Verzeichnis- oder Datendienste oder in der Geschäftsdienstschicht vorhandene Komponenten betreffen. Für die Integration von Java ES-Komponenten in solche Systeme kann die Migration von Daten oder Anwendungscode notwendig sein.

Die J2EE-Plattform bietet ein Connector Framework, mit dem Sie vorhandene Anwendungen in die Anwendungsserver-Umgebung aufnehmen, indem Sie J2EE-Ressourcenadapter entwickeln. Message Queue bietet für die Integration diverser Anwendungen eine robuste, asynchrone Messaging-Funktion.

Testen von Prototypen und Piloten

Je nach Anpassungsgrad oder Entwicklungsaufwand müssen Sie Ihre Bereitstellungsarchitektur zu einem bestimmten Zeitpunkt wie folgt testen: Sie müssen die Lösung anhand der Anwendungsfälle testen und überprüfen, ob sie die Dienstqualitätsanforderungen erfüllt.

Wenn Sie relativ wenig kundenspezifisch entwickelte Dienste einsetzen (die Bereitstellung also überwiegend Out-of-the-Box erfolgt), müssen für Ihre Lösung lediglich die Java ES-Komponenten angepasst und ein Pilottest des Systems durchgeführt werden.

Wenn Sie jedoch eine wichtige neue Anwendungslogik entwickelt und benutzerdefinierte Dienste erstellt haben, müssen Sie wahrscheinlich umfangreichere Tests durchführen, wie Prototyp-Tests, Integrationstest usw.

Wenn diese Tests Schwachstellen Ihrer Bereitstellungsarchitektur aufzeigen, müssen Sie die Architektur ändern und erneut testen. Dieser schrittweise Prozess sollte schließlich zu einer Bereitstellungsarchitektur und einer Implementierung führen, die die Bereitstellung in einer Produktionsumgebung gewährleisten kann.

Produktionseinsatz

Der Produktionseinsatz besteht in der Anwendung Ihrer Bereitstellungsimplementierung in einer Produktionsumgebung. Diese Phase umfasst das Installieren, Konfigurieren und Starten der verteilten Anwendungen und Infrastrukturdienste in einer Produktionsumgebung, die Einrichtung von Produktionssystem-Endbenutzern sowie die Einrichtung von Single Sign-On, Zugriffsrichtlinien usw. Üblicherweise beginnen Sie mit einer eingeschränkten Bereitstellung und dehnen diese dann auf das gesamte Unternehmen aus. Hierbei führen Sie Testläufe aus, in denen Sie die Belastung stetig steigern, um zu bestätigen, dass die Qualitätsanforderungen erfüllt werden.

Bereitstellungsnachbereitung

In der Lebenszyklusphase Bereitstellungs- nachbereitung führen Sie eine bereitgestellte Lösung in einer Produktionsumgebung aus. Die Betriebsphase des Lebenszyklus beinhaltet die folgenden Aufgaben:

- **Überwachung:** Zu dieser Aufgabe gehört die regelmäßige Überwachung der Systemleistung und der Systemfunktionen.
- **Wartung:** Zu dieser Aufgabe gehören die täglich anfallenden Verwaltungsfunktionen, wie das Hinzufügen neuer Benutzer zu einem System, das Ändern von Passwörtern, das Hinzufügen neuer administrativer Benutzer, das Ändern von Zugriffsberechtigungen, die Durchführung regelmäßiger Sicherungen usw.
- **Leistungsoptimierung:** Zu dieser Aufgabe gehört die Auswertung regelmäßiger Überwachungsinformationen, um Engpässe beim Systembetrieb zu erkennen und diese dann durch Änderungen der Konfigurationseinstellungen, Hinzufügen von Kapazität usw. zu beseitigen.
- **Systemerweiterungen und Aktualisierungen:** Zu diesen Aufgaben gehört das Hinzufügen neuer Java ES-Komponenten zu einem System, um neue Funktionen verfügbar zu machen, oder das Ersetzen von Nicht-Java ES-Komponenten. Diese Änderungen erfordern unter Umständen eine Neukonzeption des Systems, die mit den ersten Phasen des Lösungslebenszyklus anfängt. Aktualisierungsaufgaben sind weniger umfangreich, da es sich meist um die Aktualisierung von Java ES-Komponenten handelt.

In diesem Kapitel enthaltene Schlüsselbegriffe

In diesem Abschnitt werden die wichtigsten in diesem Kapitel verwendeten technischen Bedingungen erläutert, wobei auf die Verwendung dieser Bedingungen im Zusammenhang mit Java ES besonders eingegangen wird.

Bereitstellung Ein Abschnitt im Lebenszyklus einer Java ES-Lösung, in dem ein Bereitstellungsszenario in ein Bereitstellungskonzept übertragen, dann implementiert, als Prototyp getestet und schließlich in einer Produktionsumgebung eingesetzt wird. Das Endprodukt dieses Prozesses wird ebenfalls als Bereitstellung (oder bereitgestellte Lösung) bezeichnet.

Bereitstellungs-szenario Eine Logische Architektur für eine Java ES-Lösung und die Dienstqualitätsanforderungen, die von der Lösung erfüllt werden müssen, um den Geschäftsanforderungen zu entsprechen. Die Dienstqualitätsanforderungen betreffen unter anderem die Leistung, Verfügbarkeit, Sicherheit, Zweckmäßigkeit und Skalierbarkeit oder die latente Kapazität. Ein Bereitstellungsszenario ist der Ausgangspunkt für ein Bereitstellungskonzept.

Entwicklung Eine Aufgabe im Bereitstellungsprozess der Java ES-Lösung, bei der die angepassten Komponenten einer Bereitstellungs- architektur programmiert und getestet werden.

Bereitstellungs-vorbereitung Ein Abschnitt im Lebenszyklus der Java ES-Lösung, in dem die Geschäftsanforderungen in ein Bereitstellungs- szenario umgesetzt werden: Eine Logische Architektur und eine Reihe von Dienstqualitätsanforderungen, die die Lösung erfüllen muss.

Bereitstellungs-nachbereitung Ein Abschnitt im Lebenszyklus einer Java ES-Lösung, in dem verteilte Anwendungen gestartet, überwacht, zur Optimierung der Leistung angepasst und dynamisch mit neuen Funktionen aufgerüstet werden.

Referenz-Bereitstellungs-architektur Eine Bereitstellungs- architektur, die hinsichtlich der Leistung entworfen, implementiert und getestet wurde. Referenz-Bereitstellungsarchitekturen dienen als Ausgangspunkt für die Konzipierung von Bereitstellungsarchitekturen für individuell angepasste Lösungen.

Anwendungsfall Eine bestimmte Endbenutzeraufgabe oder eine Reihe von Aufgaben, die eine Verteilte Unternehmensanwendung ausführt und die als Basis für die Konzipierung, das Testen und das Messen der Leistung der Anwendung dient.

◆ ◆ ◆ ANHANG A

Java ES-Komponenten

Java ES besteht aus einer Sammlung von Produktkomponenten und gemeinsam genutzten Komponenten, die interaktiv zusammenwirken, um über ein Netzwerk verteilte Anwendungen zu unterstützen. Während der Installation werden vom Java ES-Installationsprogramm auswählbare Komponenten angezeigt, von denen viele über auswählbare Unterkomponenten verfügen. Diese Komponenten und Unterkomponenten werden in diesem Anhang aufgeführt.

Dieser Anhang enthält Kurzbeschreibungen von Java ES-Komponenten und soll als Übersicht dienen. Detaillierte Informationen über spezifische Komponenten erhalten Sie in den jeweiligen Komponentendokumentationssätzen, die unter `http://docs.sun.com/app/docs/prod/entsys.5` verfügbar sind. Eine Vielzahl an Java ES-Informationen und -Ressourcen sind auch unter `http://www.sun.com/bigadmin/hubs/javaes/` verfügbar.

Die in diesem Anhang aufgeführten Java ES-Komponenten sind nach Kategorie gruppiert und werden in den folgenden Abschnitten beschrieben:

- „Systemdienstkomponenten" auf Seite 71
- „Dienstqualitätskomponenten" auf Seite 76
- „Freigegebene Komponenten" auf Seite 80

Systemdienstkomponenten

Java ES-Systemdienstkomponenten bieten die Infrastrukturdienste, die für die Unterstützung verteilter Unternehmensanwendungen benötigt werden. Wie unter „Warum Sie Java ES benötigen" auf Seite 17 beschrieben, beinhalten diese Dienste Zugangsdienste, Identitäts- und Sicherheitsdienste, Web- und Anwendungsdienste sowie Verfügbarkeitsdienste.
Java ES-Systemdienstkomponenten werden in den folgenden Abschnitten beschrieben:

- „Access Manager 7.1" auf Seite 72
- „Anwendungsserver Enterprise Edition 8.2" auf Seite 73
- „Directory Server Enterprise Edition 6.0 " auf Seite 73

Access Manager 7.1

Sun Java System Access Manager (Access Manager) integriert Authentifizierungs- und Autorisierungsdienste, Richtlinienagenten und die Identitätsföderierung, um eine umfassende Lösung zum Schutz von Netzwerkressourcen bereitzustellen. Access Manager verhindert unberechtigte Zugriffe auf Webdienstanwendungen und Webinhalte und liefert Organisationen eine Infrastruktur zur Verwaltung von digitalen Identitäten für Kunden, Mitarbeiter und Partner, die ihre webbasierten Dienste und Nicht-Webanwendung verwenden. Da diese Ressourcen möglicherweise über interne und externe Computernetzwerke hinweg verstreut sind, werden Attribute, Richtlinien und Berechtigungen definiert und auf jede Identität angewendet, um den Zugriff auf diese Technologien zu verwalten.

Access Manager beinhaltet die folgenden Unterkomponenten:

- **Access Manager Core Services:** Mit dieser Unterkomponente werden Benutzeridentitäten erstellt und verwaltet sowie Richtlinien definiert und ausgewertet, die basierend auf der Identität der Benutzer Zugriff auf die Java ES-Ressourcen bieten.
- **Access Manager Administration Console:** Fasst Identitätsdienste und Richtlinienverwaltung zusammen und liefert eine einzige grafische Schnittstelle, die Benutzern das Erstellen und Verwalten von Benutzerkonten, Dienstattributen und Zugangsregeln im Directory Server ermöglicht.
- **Common Domain Services for Federation Management:** Ermöglicht Benutzern die Verwendung einer einzelnen Identität, um auf Anwendungen zuzugreifen, die von mehreren angegliederten Dienstanbietern angeboten werden.
- **Access Manager-SDK:** Bietet eine Remote-Schnittstelle für Access Manager. Diese Unterkomponente muss auf jedem Computer installiert werden, der als Host für eine Java ES-Komponente dient, auf die Access Manager remote zugreift.
- **Access Manager Distributed Authentication User Interface:** Bietet eine Benutzerschnittstelle, mit der ein Richtlinienagent oder eine Anwendung, die in einem ungeschützten Bereich verwendet wird, mit dem in einem geschützten Bereich der Bereitstellung installierten Authentifizierungsdienst des Access Manager kommunizieren kann.
- **Access Manager Client-SDK:** Ermöglicht es dem Benutzer, eigenständige Anwendungen zu implementieren, die auf einen Access Manager-Server zugreifen können, um Dienste wie beispielsweise Authentifizierung, Single Sign-On, Autorisierung, Überwachung, Protokollierung und Security Assertion Markup Language (SAML) zu verwenden.

- **Access Manager Session Failover Client:** Erforderlich, um Access Manager-Sitzungs-Failover zu konfigurieren.

Anwendungsserver Enterprise Edition 8.2

Sun Java System Anwendungsserver (Anwendungsserver) bietet eine J2EE-kompatible Plattform für die Entwicklung und Bereitstellung von serverseitigen Java-Anwendungen und Webdiensten. Hauptfunktionen sind beispielsweise die Transaktionsverwaltung, Container-verwaltete Fortdauer, Webdienstleistung, Clustering, Hochverfügbarkeitssitzungsstatus, Sicherheit und Integrationsfähigkeiten.

Application Server beinhaltet die folgenden Unterkomponenten:

- **Domain Administration Server.** Bietet serverseitige Verwaltungsfunktionen, wie die Verwaltung und Konfiguration von Anwendungsserver und die Bereitstellung von J2EE-Komponenten und -Anwendungen.
- **Application Server-Knotenagent.** Ein leichtgewichtiger Prozess, der auf jedem Computer ausgeführt wird, der als Host für Serverinstanzen dient und eine Anzahl an Administrationsaufgaben durchführt, wie beispielsweise das Anhalten, Starten und Neustarten von Serverinstanzen.
- **Command Line Administration Tool.** Bietet befehlszeilengesteuerte Verwaltungsclients für die Verwaltung und Konfiguration von Anwendungsserver-Installationen und gehosteten Anwendungen. Das Tool hilft auch bei der Bereitstellung von Anwendungen.
- **Lastenausgleichs-Plug-In:** Wird verwendet, um die Arbeitslast zwischen mehreren (eigenständigen oder geclusterten) Application Server-Instanzen gleichmäßig aufzuteilen und dadurch den Gesamtdurchsatz des Systems zu erhöhen. Wird darüber hinaus verwendet, um Anforderungen von einer Serverinstanz zur nächsten zu aktivieren.
- **Beispielanwendungen:** Werden im Rahmen der vollständigen Application Server-Installation installiert.

Directory Server Enterprise Edition 6.0

Sun Java System Directory Server (Directory Server) ist ein LDAP-basierter Directory Server, der einen zentralen Verzeichnisdienst für Ihre Intranet-, Netzwerk- und Extranet-Informationen bereitstellt. Directory Server wird in bestehende Systeme integriert und fungiert als zentraler Ablageort für die Konsolidierung von Mitarbeiter-, Kunden-, Lieferanten- und Partnerinformationen. Sie können Directory Server dahingehend erweitern, dass die Verwaltung von Benutzerprofilen und Voreinstellungen sowie die Extranet-Benutzerauthentifizierung ermöglicht werden.

Directory Server beinhaltet die folgenden Unterkomponenten:

- **Directory Server 6 Core Server:** Bietet eine skalierbare, sichere und flexible Möglichkeit, um Identitätsdaten zu speichern und zu verwalten.
- **Directory Service Control Center:** Liefert eine browserbasierte Administrationsschnittstelle zur Konfiguration von Directory und Directory Proxy-Diensten.
- **Directory Server-Befehlszeilendienstprogramm:** Hiermit können Sie Administrationsaufgaben über die Befehlszeile durchführen.
- **Directory Proxy Server 6 Core Server:** Verbessert die Sicherheit durch Bereitstellen von virtuellen Verzeichnisfähigkeiten und durch eine erhöhte Verfügbarkeit und Skalierbarkeit von Verzeichnisdiensten.

Java DB 10.1

Java DB stellt eine leichtgewichtige Datenbank für die Java-Anwendungsentwicklung zur Verfügung. Java DB ist die unterstützte Verteilung der Open-Source-Datenbank Apache Derby mit 100 % Java-Technologie. Java ES 5 ist die erste Version, in der Java DB als Produktkomponente enthalten ist. Java DB wurde zunächst als freigegebene Komponente mit dem Namen Derby Database veröffentlicht und wurde in Java ES 2005Q4 integriert.

Java DB beinhaltet die folgenden Unterkomponenten:

- Java DB Client
- Java DB Server

Message Queue 3.7 UR 1

Sun Java System Message Queue (Message Queue) ist eine auf Standards basierende Lösung für die häufig problematische Kommunikation zwischen Anwendungen und verlässlicher Nachrichtenweiterleitung. Message Queue ist ein Messaging-System für Unternehmen, das den Java Message Service (JMS) Open Standard implementiert.

Message Queue ist jedoch nicht nur ein JMS-Anbieter, sondern verfügt auch über Funktionen, die über die Mindestanforderungen der JMS-Spezifikation hinausgehen. Über Message Queue-Software können Vorgänge, die auf unterschiedlichen Plattformen und unter unterschiedlichen Betriebssystemen ausgeführt werden, eine Verbindung mit einem allgemeinen Message Queue-Dienst herstellen, um Informationen zu senden und zu empfangen. Anwendungsentwickler können sich also auf die Geschäftslogik ihrer Anwendungen konzentrieren und müssen sich nicht mit den wenig interessanten Details der netzwerkübergreifenden Anwendungskommunikation befassen.

Das Java ES-Installationsprogramm stellt Message Queue als installierbare Einzelkomponente bereit.

Portal Server 7.1

Sun Java System Portal Server (Portal Server) ist eine identitätsbasierte Zugangsserver-Lösung. Portal Server vereint Zugangsdienste, wie Personalisierung, Aggregation, Sicherheit, Integration und Suche.

Das Java ES-Installationsprogramm stellt Portal Server als installierbare Einzelkomponente bereit.

Service Registry 3.1

Sun Java System Service Registry (Service Registry) ist ein Repository, das sowohl als Webdienst (UDDI)-Registrierung als auch als Enterprise Business XML (ebXML)-Registrierung für die Unterstützung von Service-Oriented Architecture (SOA)-Anwendungen dient. Die UDDI-Registrierung wird für die Registrierung und das Auffinden von Webdiensten verwendet und die ebXML-Registrierung für das Speichern und Verwalten der Informationsartefakte, die für die Unterstützung der Unternehmensprozessintegration erforderlich sind. Zu diesen Artefakten gehören Metadaten, wie XML-Schemata, Unternehmensprozessregeln, Webdienstzugriffssteuerung, Versionskontrolle, Klassifizierungsschemata usw.

Service Registry beinhaltet die folgenden Unterkomponenten:

- Service Registry Client Support
- Service Registry Deployment Support

Web Server 7.0

Sun Java System Web Server (Web Server) ist ein sicherer, auf Industrierstandards basierender Multiprozess- und Multithread-Webserver. Web Server bietet mittleren bis großen Unternehmen hohe Leistung, Zuverlässigkeit, Skalierbarkeit sowie Verwaltungsfunktionen.

Web Server beinhaltet die folgenden Unterkomponenten:

- Web Server CLI
- Web Server Core
- Web Server-Beispiele

Dienstqualitätskomponenten

Java ES-Dienstqualitätskomponenten verbessern die Dienstqualität von Systemdienstkomponenten oder verteilten Anwendungskomponenten. Bei einigen handelt es sich um Verfügbarkeitskomponenten, die für einen nahezu kontinuierlichen Betrieb des Systems eingesetzt werden. Bei anderen handelt es sich um Zugriffskomponenten, die den gesicherten Zugriff von Endbenutzern auf Systemdienste unterstützen. Oder es handelt sich um Systemverwaltungskomponenten, mit denen die Zweckmäßigkeit der Java ES-Lösungen verbessert wird.

Die Komponenten, durch die Java ES-Dienstkomponenten unterstützt werden, werden in die folgenden Kategorien eingeteilt und in diesem Abschnitt beschrieben:

- „Verfügbarkeitskomponenten" auf Seite 76
- „Zugriffskomponenten" auf Seite 78
- „Überwachungskomponenten" auf Seite 79

Verfügbarkeitskomponenten

Verfügbarkeitskomponenten sorgen für eine nahezu kontinuierliche Systembetriebszeit der Systemdienstkomponenten und Anwendungskomponenten. In diesem Abschnitt werden die folgenden Java ES-Verfügbarkeitskomponenten beschrieben:

- „High Availability Session Store 4.4.3" auf Seite 76
- „Sun Cluster 3.1 8/05 und Sun Cluster Agents 3.1" auf Seite 77
- „Sun Cluster Geographic Edition 3.1 2006Q4" auf Seite 78

High Availability Session Store 4.4.3

Sun Java System High Availability Session Store (HADB) bietet einen Datenspeicher, mit dem Anwendungsdaten selbst bei einem Ausfall verfügbar gemacht werden können. Diese Möglichkeit ist besonders wichtig, um die einer Client-Sitzung zugeordneten Statusinformationen wiederherzustellen. Wenn während einer Sitzung ein Fehler auftritt, müssen in der neu erstellten Sitzung alle Vorgänge noch einmal wiederholt werden.

Die folgenden Java ES-Komponenten bieten Dienste, die Sitzungsstatusinformationen speichern: Anwendungsserver, Access Manager und Message Queue. Anwendungsserver ist jedoch die einzige Komponente, die HADB-Dienste nutzen kann, um während eines Ausfalls den Sitzungsstatus zu erhalten.

Das Java ES-Installationsprogramm stellt HADB als installierbare Einzelkomponente zur Verfügung. Für die HADB-Dienste werden jedoch sowohl eine Server- als auch eine Client-Unterkomponente benötigt.

Sun Cluster 3.1 8/05 und Sun Cluster Agents 3.1

Hinweis – Sun Cluster-Komponenten werden nur auf der Solaris-Plattform unterstützt.

Sun Cluster-Software bietet Hochverfügbarkeitsdienste und Skalierbarkeit für Java ES sowie für auf der Java ES-Infrastruktur basierende Anwendungen.

Ein Cluster besteht aus lose miteinander verbundenen Computern (Cluster-Knoten), die zusammen eine einzelne Client-Ansicht der Dienste, Systemressourcen und Daten bieten. Intern verwendet der Cluster redundante Computer, Interconnects, Datenspeicher und Netzwerkschnittstellen zur Bereitstellung der Hochverfügbarkeit für clusterbasierte Dienste und Daten. Sun Cluster-Software überwacht fortlaufend den Zustand der Mitgliedsknoten und anderer Clusterressourcen und verwendet die interne Redundanz zur Bereitstellung eines beinahe unterbrechungsfreien Zugriffs auf diese Ressourcen, selbst wenn Fehler auftreten.

Das Java ES-Installationsprogramm stellt die Sun Cluster-Core-Unterkomponente und Sun Cluster-Agenten als installierbare Einzelkomponenten bereit. Die folgenden Sun Cluster-Agenten sind in Java Enterprise System enthalten:

Hinweis – In der folgenden Liste steht *HA* für *High Availability (hohe Verfügbarkeit)*.

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

Hinweis – Die Liste der Agenten ist bei SPARC und x86 unterschiedlich. Detaillierte Informationen zu Sun Cluster-Agenten erhalten Sie in der Sun Cluster-Dokumentation unter `http://docs.sun.com/app/docs/prod/entsys.5`.

Sun Cluster Geographic Edition 3.1 2006Q4

Sun Cluster Geographic Edition ist eine geschichtete Erweiterung der Sun Cluster-Software. Diese Erweiterung schützt Anwendungen mithilfe von mehreren geografisch getrennten Clustern und einer redundanten Infrastruktur, durch die Daten zwischen diesen Clustern repliziert werden, vor unerwarteten Unterbrechungen. Java ES 5 ist die erste Version, die Sun Cluster Geographic Edition als Java ES-Produktkomponente enthält.

Sun Cluster Geographic Edition beinhaltet die folgenden Unterkomponenten:

- Sun Cluster Geographic Edition Core
- Sun StorEdge Availability Suite
- Hitachi Truecopy Data Replication Support (nur SPARC)
- EMC SRDF Data Replication

Hinweis – Sun Cluster Geographic Edition wird nicht auf Solaris x86 unterstützt.

Zugriffskomponenten

Zugriffskomponenten bieten Front-End-Zugriff auf Systemdienste, der häufig über Internetstandorte erfolgt, die sich außerhalb der Firewall des Unternehmens befinden. In diesem Abschnitt werden die folgenden Java ES-Zugriffskomponenten beschrieben:

- „Portal Server, Secure Remote Access 7.1" auf Seite 78
- „Web Proxy Server 4.0.4" auf Seite 79

Portal Server, Secure Remote Access 7.1

Sun Java System Portal Server, Secure Remote Access (Portal Server, Secure Remote Access) ist eine Erweiterung für Portal Server und bietet browserbasierten sicheren Remote-Zugriff auf Portal Server-Content und -Dienste von einem beliebigen Remote-Browser aus. Dadurch wird der Einsatz von Client-Software überflüssig. Durch die Integration in Portal Server wird gewährleistet, dass die Benutzer auf sichere Weise auf den Content und die Dienste zugreifen können, für die sie zugriffsberechtigt sind.

Portal Server Secure Remote Access beinhaltet die folgenden Unterkomponenten:

- **Portal Server Secure Remote Access Core.** Liefert die Core-Funktionalität.

- **Gateway:** Liefert eine Schnittstelle und Sicherheitsbarriere zwischen Remote-Benutzer-Sitzungen, die aus dem Internet kommen, und dem Intranet Ihres Unternehmens. Das Gateway präsentiert in sicherer Weise Inhalte aus internen Webservern und Anwendungsservern über eine einzige Schnittstelle mit einem Remote-Benutzer und steuert die Kommunikation zwischen dem Portal Server und den verschiedenen Gateway-Instanzen.
- **Netlet Proxy:** Ermöglicht dem Benutzer die sichere Ausführung gängiger TCP/IP-Dienste über das Internet und andere nicht sichere Netzwerke. Mit Netlet können Sie Anwendungen wie Telnet, SMTP, HTTP und Anwendungen mit festem Anschluss ausführen. Netlet ermöglicht Remote-Zugriffe und -Betrieb von Dateisystemen und Verzeichnissen und stellt eine sichere Kommunikation zwischen dem Netlet-Applet auf dem Client-Browser, dem Gateway und dem Anwendungsserver sicher.
- **Rewriter Proxy:** Ermöglicht sichere HTTP-Verbindungen zwischen Gateway und Intranet. Rewriter ermöglicht den sicheren Zugriff auf Webseiten von Unternehmens-Intranets von außerhalb des Intranets durch Umwandlung von Weblinks und das Erstellen von Regelsätzen für den Umgang mit Internet-Webseiten.

Web Proxy Server 4.0.4

Sun Java System Web Proxy Server (Web Proxy Server) bietet Cache-, Filter- und Verteilungsfunktionen für Webinhalte. Web Proxy Server wird häufig innerhalb von Firewalls eingesetzt, um die Anzahl der Anforderungen für Remote-Content Server zu reduzieren; außerhalb von Firewalls bietet Web Proxy Server einen sicheren Gateway für eingehende Internetanforderungen.

Das Java ES-Installationsprogramm stellt Web Proxy Server als installierbare Einzelkomponente zur Verfügung.

Überwachungskomponenten

Sun Java System Monitoring Console 1.0 (Monitoring Console) beinhaltet einen Master-Agenten, durch den sämtliche in einer Java ES-Bereitstellung enthaltenen Knotenagenten miteinander verbunden werden. Die Monitoring Console wird von Sun Java System Monitoring Framework 2.0 (Monitoring Framework) unterstützt, einer freigegebenen Komponente, die die Ausstattung und die Knotenagenten, die von jeder überwachten Komponente zur Veröffentlichung ihrer Attribute zur Überwachung benötigt werden, bereitstellt. Jede Produktkomponente veröffentlicht die Objekte, die ihre zu überwachenden Attribute darstellen, und ein Knotenagent koordiniert eine Ansicht mehrerer Komponenten auf einem Hostcomputer. Detaillierte Informationen über die Überwachung erhalten Sie im *Sun Java Enterprise System 5 Überwachungshandbuch*.

Freigegebene Komponenten

Gemeinsam genutzte Komponenten bieten die lokale Unterstützung für Dienste und Technologien, von denen die Systemdienstkomponenten und Dienstqualitätskomponenten von Java ES abhängen. Diese Komponenten sind lokale Bibliotheken, die von jeder Java ES-Komponente, die auf bestimmten Hostcomputern ausgeführt wird, freigegeben werden können. Das Java ES-Installationsprogramm installiert automatisch die freigegebenen Komponenten, die für die Unterstützung anderer auf einem Hostcomputer installierten Java ES-Komponenten erforderlich sind.

Java ES beinhaltet die folgenden freigegebenen Komponenten:

- ACL (Apache Common Logging) 1.0.4
- ANT (auf Jakarta ANT Java/XML basierendes Tool) 1.6.5
- BDB (Berkeley Database) 4.2.52
- Common Agent Container 1.1 (nur Sun Cluster)
- Common Agent Container 2.0
- FastInfoSet 1.0.2
- ICU 3 (International Components for Unicode) 3.2
- J2SE (Java 2 Platform, Standard Edition) 5.0 Update 6 (Version 5.0 Update 3 wird bei HP-UX unterstützt)
- JAF (JavaBeans™ Activation Framework) 1.0.3
- JATO (Java Studio Web Application Framework) 2.1.5
- JavaHelp™ 2.0
- JavaMail™ API 1.3.2
- JAXB (Java Architecture for XML Binding) 2.0.3
- JAXP (Java API for XML Processing) 1.3.1
- JAXR (Java API for XML Registries) 1.0.8
- JAXRPC (Java API for XML-based Remote Procedure Call) 1.1.3_01
- JAXWS (Java API for Web Services) 2.0
- JDMK (Java Dynamic Management Kit) 5.1.2
- JSS (Java Security Services) 4.2.4
- JSS3 (Network Security Services for Java) 3.1.11
- JSTL (JavaServer Pages™ Standard Tag Library) 1.0.6
- KTSE (KT Search Engine) 1.3.4
- LDAP C SDK 6.0
- LDAP Java SDK 4.19

- MA Core (Mobile Access Core) 6.3.1
- NSPR (Netscape Portable Runtime) 4.6.3
- NSS (Network Security Services) 3.11
- NSSU (Network Security Service Utilities) 3.11
- SAAJ (SOAP mit Attachments-API für Java) 1.3
- SASL (Simple Authentication and Security Layer) 2.19
- Sun Explorer Data Collector (nur Solaris OS) 4.3.1
- Sun Java System Monitoring Framework 2.0 (unterstützt Monitoring Console 1.0)
- Sun Java Web Console 3.0.2
- WSCL (Web Services Common Library) 2.0
- XWSS (XML Web Services Security) 2.0

Index

C

D

E

U

V

W

Z