



WHITE PAPER • MÄRZ 2018



Die wichtigsten Anforderungen an eine Lösung für Job-Scheduling und Workload Automation

Die herkömmliche Planung von Batch-Jobs stößt an ihre Grenzen

Inhaltsverzeichnis

Kurzzusammenfassung	3
Section 1	4
Job-Scheduling	
Section 2	4
Integrierter Managed File Transfer	
Section 3	5
Zentrale und skalierbare Multi-Tenancy-Architektur	
Section 4	5
Schnelle, flexible Implementierungen dank Objektorientierung	
Section 5	5
Native Unterstützung von Applikationen	
Section 6	6
Dynamische, ereignisbasierte Automatisierung	
Section 7	7
Bedingte Geschäftsregeln	
Section 8	7
Workload Balancing	
Section 9	8
Self-Service-Management	
Section 10	8
Datensicherheit und Compliance	
Section 11	9
Fazit	

Kurzzusammenfassung

Die Herausforderung

Herkömmliches Batch-Job-Scheduling findet zwar weiterhin in vielen Unternehmen Anwendung, die Anforderungen sind allerdings aufgrund von unterschiedlichen IT-Plattformen und -umgebungen sowie rund um die Uhr ausgeführten Geschäftsprozessen gestiegen. Zudem sind häufig komplexe Abhängigkeiten zwischen den unterschiedlichen Prozessen und eine dynamische Steuerung für den Start der Jobbearbeitung auf Basis der sich ändernden Bedingungen zu berücksichtigen. Die aktuellen komplexen Job-Scheduling-Anforderungen machen daher eine umfassende Lösung für die Workload-Automatisierung erforderlich.

Die Herausforderung

Über welche Leistungsmerkmale muss eine Unternehmenslösung für Job-Scheduling und Workload Automation heute verfügen? In den vergangenen 15 Jahren haben wir die wichtigsten Anforderungen globaler Unternehmen zusammengestellt. Alle haben erfolgreich die Kernprozesse in den Rechenzentren automatisiert, um ihre wichtigsten Unternehmensapplikationen und Dienste zu unterstützen.

Das Ergebnis

So konnten sie eine Verkürzung der Verarbeitungszeit um 70 Prozent sowie die Vermeidung von 90 Prozent der IT-Prozessfehler erreichen und gleichzeitig den manuellen Aufwand um 90 Prozent reduzieren. Auch Sie können das erreichen. Wir zeigen Ihnen wie.

SECTION 1

Job-Scheduling

Auch wenn die Anforderungen an Job-Scheduler mit der Zeit gestiegen sind, müssen diese immer noch eine wesentliche Funktion erfüllen: ein effektives Job-Scheduling gewährleisten. Einfach ausgedrückt handelt es sich beim Job-Scheduling um die zuverlässige, sequentielle Ausführung von Batch-Programmen. Um diese Aufgabe ordnungsgemäß zu erfüllen, muss ein Job Scheduler über eine Vielzahl an Merkmalen verfügen, die eine effektive Planung verteilter Workloads in Übereinstimmung mit den Zielen des Geschäftsbetriebs gewährleisten.

Als Grundlage für die Planung verwenden Unternehmen meist einen oder mehrere Geschäftskalender. Diese Kalender legen wichtige Termine, z. B. für die Lohn- und Gehaltsabrechnung, Geschäftsberichte sowie Monats- und Quartalsabschlüsse fest. Das größte Problem dieser Kalender und Scheduling-Funktionen ist die fehlende Flexibilität und Zuverlässigkeit. Kann der Scheduler auf sich ändernde Geschäftsanforderungen reagieren und stets vorhersehbare Ergebnisse liefern?

In der heutigen auf die Cloud fokussierten Welt, ist eines der wichtigsten Leistungsmerkmale eine Kompatibilität der Lösung mit sämtlichen Cloud-basierten Lösungen. Dies erfordert eine enge Zusammenarbeit mit sowohl REST- als auch SOAP-basierten Frameworks. Und wenn Sie derzeit sogar noch zu den Nutzern von herkömmlichen SaaS-basierten Cloud-Lösungen zählen, ist es besonders wichtig, sich mit einem hohen Maß an Kontrolle für die Zukunft zu rüsten.

Programme und Applikationen müssen zu bestimmten Zeitpunkten in der vorgegebenen Reihenfolge mit verschiedenen Ressourcen- und Prioritätsanforderungen auf verschiedenen Plattformen laufen. Die Scheduler müssen flexibel genug sein, um diese breite Vielfalt an Anforderungen im Hinblick auf die Technologien, den Geschäftsbetrieb und die Ressourcen zu erfüllen. Wenn der Job-Scheduler in der Lage ist, die Prozesse in der erforderlichen Reihenfolge zu planen und Ressourcenkonflikte zu lösen, wird die Ausführung der Applikationen schneller und berechenbarer und der Durchsatz beim Scheduling steigt. Leider konzentrieren sich die meisten Scheduling-Lösungen auf sequenzielle Aktivitäten und ignorieren die Möglichkeiten zum Ausgleich und zur Priorisierung von Workloads.

SECTION 2

Integrierter Managed File Transfer

Ob für den Austausch von Daten zwischen unterschiedlichen Standorten, mit externen Geschäftspartnern oder zwischen internen Systemen – die meisten Workloads setzen eine sichere und schnelle Datenübertragung voraus.

Bei einer separaten nicht integrierten und unkoordinierten Übertragung handelt es sich um einen „Unmanaged“ File Transfer-Ansatz, der zu einer weiteren unzusammenhängenden „Automatisierunginsel“ führt und somit zu einem weiteren schwachen Kettenglied im Unternehmensprozess. Die Verwaltung von Datenübertragungen unterliegt meistens manueller Koordination oder zeitbasiertem Scheduling, was nicht nur ineffizient ist, sondern auch ungenaue und unvollständige Daten in den Prozess einbringt.

Eine Job-Scheduling- bzw. Workload-Automation-Lösung sollte über einen integrierten Managed File Transfer verfügen, sodass Fluss und Qualität der Daten prozessübergreifend optimiert werden. Ein voll automatisierter, koordinierter Managed File Transfer ermöglicht eine konsistente und verlässliche, durchgängige Verarbeitung. So werden Ergebnisse einfacher prognostizierbar und im Unternehmen treten weniger Fehler auf.

SECTION 3

Zentrale und skalierbare Multi-Tenancy-Architektur

Ein Job-Scheduler bzw. eine Workload-Lösung mit zentralisierter Architektur nutzt den Vorteil der Datenspeicherung in einem zentralen Datenbanksystem. Alle Schedulingdaten einschließlich der Skripte sind immer zentral verfügbar und ermöglichen eine umfassende Dokumentation vergangener, aktueller und zukünftiger Prozesse, sodass die höchst mögliche Betriebssicherheit erzielt wird. Dank zentralisierter Kontrolle wird der Wartungsaufwand auf ein absolutes Minimum reduziert und jegliche Probleme im Zusammenhang mit nicht aufeinander abgestimmten Scheduling-Regeln vermieden.

Eine Multi-Tenancy-Architektur reduziert die Hardware- und Administrationskosten und senkt die Gesamtbetriebskosten.

Weil viele Produkte immer noch auf einer einzigen Engine beruhen und nicht auf einer Active-Active-Konfiguration basieren, erlauben sie in Spitzenbelastungszeiten keine

Skalierung. Das heißt, Sie müssen neue Hardware kaufen oder einen weiteren Scheduler bzw. ein weiteres Workload-System einrichten, um den zusätzlichen Workload verarbeiten zu können.

Ein weiterer relevanter Faktor ist die Programmiersprache der Automatisierungsenge. Mit Java entwickelte Engines haben bei hohen Workloads häufig Probleme mit der Performance und erfordern zusätzliche Hardware.

Multi-Tenancy bezeichnet ein Verfahren der Softwarearchitektur, bei der eine einzige Softwareinstanz gleichzeitig mehrere Kundenunternehmen (Mandanten) bedient. Bei einer Multi-Tenancy-Architektur partitioniert die Softwareapplikation ihre Daten und Konfiguration für die einzelnen Mandanten, sodass jedes Unternehmen mit einer kundenspezifischen logischen Instanz dieser Applikation arbeitet. Weil man die IT heute als Dienstleistung betrachtet, ist es wichtig, eine logische Trennung zwischen verschiedenen Kunden bzw. Unternehmen herbeiführen zu können, weil dies das Reporting und die Rückberechnung erbrachter Dienstleistungen an die entsprechende Abteilung vereinfacht. Diese Trennung erhöht auch die Sicherheit.

SECTION 4

Schnelle, flexible Implementierungen dank Objektorientierung

Ein objektorientiertes Design besitzt ein enormes Zeiteinsparungspotenzial, weil Sie damit eine einzige generische Job-Definition für Jobs mit unterschiedlichen Parametern auf hunderten verschiedenen Zielsystemen verwenden können. Dieses Verfahren beruht auf dem „Modularkonzept“. Beispiel: Stellen Sie sich einen Backup-Prozess vor, der auf allen UNIX®-Servern Ihres Rechenzentrums einmal pro Woche als volles Backup und jeden zweiten Tag als inkrementelles Backup laufen muss.

Bei Lösungen ohne Objektorientierung würden Sie hunderte Job-Definitionen dafür benötigen. Außerdem würde dieser Ansatz zu einem echten Wartungsapptraum führen, da sämtliche Änderungen für hunderte einzelner Job-Definitionen vorgenommen werden müssten – und die wichtigste würden Sie womöglich auch noch übersehen.

Im Gegensatz dazu beschleunigt die Objektorientierung die Implementierung und schafft eine flexible Plattform, die schnelle Änderungen in der IT-Umgebung ermöglicht.

SECTION 5

Native Unterstützung von Applikationen

Ob es sich nun um selbst entwickelte oder gekaufte Applikationen handelt – ein Job-Scheduling ist für alle unerlässlich. In der Vergangenheit wurde beim Job-Scheduling und Systemmanagement immer das Management der verschiedenen IT-Infrastrukturkomponenten in den Vordergrund gestellt. Die Netzwerke, CPUs, Datenbanken, Controller, Programme, Festplatten und weitere Komponenten erfordern Management, Überwachung und Scheduling. Im Laufe der Zeit wurde den IT-Experten klar, dass die

Fokussierung auf eine bestimmte Komponente ohne Berücksichtigung der Applikationen, die diese unterstützt, nicht dem übergeordneten Ziel der Überwachung und Verwaltung der Unternehmensprozesse genügen konnte.

Die Aufgabe eines modernen Job-Schedulers besteht in der Administration der gesamten Applikation und nicht nur der unterstützenden Komponenten. In vielen IT-Abteilungen besteht der einzige Service für die anderen Unternehmensteile

in der Administration und Überwachung der geschäftlichen Kernapplikationen und der kontinuierlichen Einhaltung der Service Level Agreements auf Basis dieser Applikationen. Die meisten Applikationspakete verfügen über einen bestimmten Typ einer Application Program Interface (API), die für Programmierer und anderes technisches Personal vorgesehen ist und dem Scheduler ermöglicht, direkt mit den Applikationspaketen zu kommunizieren, um Jobs zu starten und zu überwachen.

Die von Ihnen gewählte Lösung muss für eine native Integration der zahlreichen Einzelapplikationen in Ihrer IT-Umgebung in einen umfassenden und nahtlosen Prozessfluss sorgen können. Sie sollte die Verwaltung externer Schnittstellen vereinfachen, sodass eine Applikation einen effizienten Datenaustausch mit allen Ihren anderen Applikationen gewährleisten kann. Außerdem sollte sie keinerlei kundenspezifische Programmierarbeiten und manuelle Routineüberprüfungen zwischen den Prozessschritten erfordern und Ihnen somit den entsprechenden Zeitaufwand und Fehler ersparen.

SECTION 6

Dynamische, ereignisbasierte Automatisierung

Herkömmliche Scheduler ignorieren die Applikationen als Datenquelle zur Automatisierung der Unternehmensprozesse. Dadurch verschenken sie die Möglichkeit zur Beseitigung von Latenzen und zu dynamischen Reaktionen auf veränderte Geschäftsanforderungen. Einfache Job-Scheduler können die heutigen, steigenden Anforderungen an die Automatisierung, Integration und Beschleunigung komplexer Geschäftsprozesse nicht erfüllen.

Die Applikationsdaten in Flat Files, Berichten und relationalen Datenbanken repräsentieren den aktuellen Unternehmensstatus. Der Cash Flow schwankt, die Bestände erhöhen oder verringern sich und Rechnungen werden täglich und stündlich beglichen. Um eine beschleunigte Verarbeitung der Applikationsdaten zu ermöglichen, müssen die Automatisierungslösungen auf Basis der aktuellen Unternehmensdaten dynamisch auf einen sich ändernden Unternehmensstatus reagieren.

Nehmen wir zum Beispiel an, der Bestand eines bestimmten Artikels sei unter das Nachbestellniveau gesunken. Eine normale Vorgehensweise beim Scheduling bestünde darin, in bestimmten regelmäßigen Abständen – einmal täglich oder wöchentlich – die Bestände zu ermitteln. Zur Beschleunigung dieses Prozesses müsste im Unternehmen ein Programm zur Überprüfung der Nachbestellmenge geschrieben und anschließend eine bestimmte Handlung vollzogen werden: mehr Artikel bestellen. Je nach Applikation kann dies Änderungen an verschiedenen Programmen erfordern, was zu einem erheblichen zeitlichen und finanziellen Aufwand führt.

Ein moderner Job-Scheduler muss diesen Applikationsprozess beschleunigen können. Mit geringstem Aufwand muss er die Prozessabläufe so konfigurieren, dass die sich ändernden Applikationsdaten analysiert und anschließend auf Basis dieser Daten entsprechende Ereignisse ausgelöst werden. Die Design-Tools müssen eine Abfrage und Statusprüfung der Datenwerte erlauben, um entweder sofort oder zu einem definierten späteren Zeitpunkt Ereignisse auszulösen. Dadurch wird das IT-Personal in die Lage versetzt, die Durchführung der Geschäftsprozesse ohne kostspielige Modifikationen der Quellcodes beschleunigen zu können. Die Möglichkeit zur Einbeziehung komplexer Abhängigkeiten muss ebenfalls vorhanden sein, um kundenspezifische Lösungen zu schaffen, die eine weitergehende Automatisierung und Integration der geschäftlichen Unternehmensprozesse herbeiführen. Eine dynamische Applikationsautomatisierung steht im Gegensatz zu den Möglichkeiten typischer Job-Scheduling-Tools, die lediglich eine feste zeitliche Abfolge von Systemereignissen oder Skripten steuern. Um eine intelligente Business Automation zu erzielen, die den Anforderungen eines wachsenden Unternehmens gewachsen ist, müssen die Prozesse dann ausgeführt werden, wenn der Geschäftsbetrieb diese erfordert, und nicht in vorher definierten Zeitintervallen. Aus diesem Grund ist es unerlässlich, dass Prozesse in Abhängigkeit von bestimmten Applikationsdaten oder Geschäftsereignissen gestartet werden.

SECTION 7

Bedingte Geschäftsregeln

Eine weitere wichtige Forderung an den Job-Scheduler ist die Festlegung bedingter Geschäftsregeln für die Unternehmensprozesse. Diese Regeln können auf Basis der Applikationsdaten oder auf Basis anderer Faktoren aufgestellt werden. Zur Modellierung bedingter Geschäftsregeln benötigt man leistungsfähige Tools.

Bei der Implementierung von Unternehmensregeln werden logische Operatoren, wie „größer als“, „kleiner als“, „ist gleich“, „ähnlich wie“ usw. dazu verwendet, eine Reihe verschiedener Bedingungen im Geschäftsablauf zu prüfen. Eine breite Vielfalt bedingter Operatoren zur Abfrage von Datenwerten und zur ergebnisbasierten Entscheidungsfindung ist bereitzustellen. Diese bedingten Operationen müssen sich einfach anlegen lassen und durch eine Benutzerschnittstelle mit natürlicher Sprache unterstützt werden, damit auch nichttechnische Mitarbeiter ohne Programmierung diese Geschäftsregeln anlegen können.

Einige einfache Beispiele für die Verwendung bedingter Logik sind:

„Wenn der Wert in Zeile x der Tabelle y größer als 100 ist, starte Prozess A.“

„Wenn mehr als 50 Transaktionen in Tabelle y vorhanden sind, starte Prozess A.“

Durch das Hinzufügen weiterer logischer Operatoren wie „und“ und „oder“ lassen sich komplexere Unternehmensregeln zur

Modellierung der geschäftlichen Abläufe leicht erstellen.

Diese bedingte Logik vor, während und nach der Durchführung eines Jobs anwenden zu können, ist entscheidend, weil sie bei der Modellierung komplexer Geschäftsprozesse eine enorme Flexibilität bietet. Dadurch kann jeder Prozess überprüft werden, um sicherzustellen, dass alle früheren, aktuellen und zukünftigen Bedingungen gültig sind. Häufig macht diese bedingte Logik manuelle Eingriffe und Programmierungen überflüssig. Ein einfaches Beispiel ist die Prüfung, ob eine bestimmte Aufgabe abgeschlossen ist oder die Voraussetzung für einen Programmstart erfüllt ist. Wie lange ein aktueller Job bereits läuft, um anschließend zu entscheiden, ob er bei Überschreitung der vorgesehenen Ausführungsdauer abgebrochen wird, kann ebenfalls geprüft werden. Unter Einbeziehung zahlreicher Eingangsgrößen, temporärer Bedingungen und Ereignisse ist der Aufbau weitaus komplexerer Prozesse möglich.

Mit bedingter Logik lässt sich ebenfalls eine bestmögliche Wiederverwendung von Workflows erzielen. Benötigt eine bestimmte Geschäftseinheit einen zusätzlichen Schritt innerhalb eines Prozesses, kann dieser optional angepasst werden. Das Ergebnis ist ein Standard-Workflow, der unternehmensintern mehrfach wiederverwendet werden kann und somit den Wartungsaufwand reduziert, aber auch – und das ist noch wichtiger – konsistente Prozesse in den unterschiedlichen Geschäftsbereichen gewährleistet.

SECTION 8

Workload Balancing

Die Applikationsverarbeitung lässt sich heute nicht mehr vom Konzept der Lastverteilung trennen. Eine Lastverteilung sorgt dafür, dass den konkurrierenden Aufgaben und Applikationen die knappen IT-Ressourcen nur in dem benötigten Umfang zugewiesen werden. Werden Ressourcenkonflikte ignoriert, kann es zu einer schweren Beeinträchtigung der IT und letztlich zu Service-Level-Problemen kommen. Effiziente Workload-Automation-Produkte müssen über wirkungsvolle Lösungen für das Ressourcenmanagement verfügen.

Mit intelligent ausgeglichenen Workloads werden eine effiziente Nutzung von Ressourcen und Services sowie reduzierte Administrationskosten erzielt.

Workload Balancing erlaubt Ihnen außer der Prioritätensetzung auch das Anlegen mehrerer Warteschlangen, in denen Jobs mit ähnlicher Priorität in Gruppen geordnet und gemäß einem vorher festgelegten Algorithmus durchgeführt werden. Außerdem müssen sich die Prioritäten und Warteschlangen leicht an die sich ändernden Anforderungen im Unternehmen anpassen lassen. Eine moderne Job-Scheduling- oder Workload-Automatisierungslösung muss über Funktionen für das Anlegen und Verwalten von Warteschlangen, die Änderung der Warteschlangenprioritäten sowie die Zuweisung von Prioritäten für Prozesse und Zeitpläne als Leistungsmerkmale verfügen. Diese Fähigkeiten optimieren den Datendurchsatz der Applikation und sorgen für eine gute Verteilung der Last zwischen allen Systemen und Applikationen.

SECTION 9

Self-Service-Management

Viele moderne Unternehmensapplikationen dienen der Unterstützung von Endnutzern. Diese können Dienste anfordern, eigene Berichte generieren und rund um die Uhr Anfragen an das System stellen. Dies stellt für das IT-Personal ein Problem dar, weil die Benutzeranfragen Schwankungen der Systemlast verursachen, die einen Einfluss auf die Performance des gesamten Systems haben. Diese Lastschwankungen sind oft so stark, dass das IT-Personal die vorgesehenen Service Level nicht mehr einhalten kann. Zudem kollidieren einige Benutzeranfragen mit anderen missionskritischen Geschäftsprozessen. Die Herausforderung besteht darin, den Endnutzern auch weiterhin die Freiheit zu geben, ihre Systemnutzung zu maximieren, während gleichzeitig eine vernünftige Systemlast und das Management von Ressourcenkonflikten gewährleistet sein müssen.

Genau wie die Unternehmensapplikationen selbst muss auch das Job-Scheduling- oder Workload- Automation-Produkt flexibel und benutzerfreundlich sein, damit der Endnutzer die Funktionen des Produkts voll ausschöpfen kann. Die Endnutzer wissen, wie ihre Geschäftsprozesse beschleunigt oder optimiert werden können. Deshalb ist es wichtig, einen Self-Service vorzusehen, der den Endnutzern die Anforderung und Überwachung von Jobs oder Prozessen bis hin zu deren Fertigstellung erlaubt.

Der Schlüssel zur Umsetzung all dieser Anforderungen im Unternehmen liegt in der Fähigkeit, die richtigen Verarbeitungs- und Priorisierungsgrundsätze zur Behandlung aller von den Nutzern angeforderten Jobs festzulegen. Das IT-Personal kann Prozesswarteschlangen, Prioritäten und

Endnutzerprofile anlegen und pflegen. Zur Vermeidung unnötiger Systemlasten und Ressourcenengpässe können diese Parameter anschließend genauer definiert werden. Warteschlangen können so angelegt werden, dass während bestimmter Tageszeiten nur bestimmte Klassen von Serviceanfragen bearbeitet und/oder ausgeführt werden. Dadurch können Endnutzer ihre Jobanforderungen flexibel und ungeachtet der tatsächlichen Ausführungszeit abschicken, ohne dass hierfür das IT-Personal involviert werden muss. Die Endnutzer verfügen so aus ihrer Sicht über eine erstklassige Kontrolle und Übersicht. Start, Überwachung und Abbruch von Anfragen können ohne Eingriffe des IT-Personals durchgeführt werden. Damit müssen IT-Mitarbeiter keine unnötige Zeit mehr mit der Bearbeitung von Ad-hoc-Anforderungen von Nutzern über Telefon oder E-Mail verschwenden. Außerdem können keine kostspieligen Fehler oder Missverständnisse bei der Übermittlung der Serviceanforderung an die IT-Abteilung mehr auftreten. Zudem werden auch einfache aber kostspielige Fehler bei der Bearbeitung von Ad-hoc-Serviceanforderungen aufgrund fehlerhafter Daten vermieden, weil die Eingangsdaten direkt zum Zeitpunkt der Nutzeranforderung validiert werden.

Additionally, in today's fast-paced world, it is essential to turn data into information that can be placed into the hands of users. Because our workload automation has all the data about our processing position and how it impacts the business, this can be turned into actionable information and presented to users in dashboards. These should augment the information about process states with real business data to create meaningful information.

SECTION 10

Datensicherheit und Compliance

Wenn die Programme für die Lohn- und Gehaltsabrechnung oder den Zahlungsverkehr gemeinsam mit ihren Datenausgaben auf demselben Server gespeichert werden, kann sich jeder Administrator oder Hacker unbefugt und unbemerkt Zugang zu diesen virtuellen Maschinen verschaffen und Änderungen vornehmen. Diese Situation stellt ein hohes Sicherheitsrisiko dar und kann drastische Folgen für Ihren Geschäftsbetrieb haben.

Um dieses Risiko zu vermeiden, müssen Sie die folgenden Punkte beachten:

- Alle Programme und Programmcodes werden in einer zentralen Datenbank gespeichert und verschlüsselt, auf die nur einige wenige autorisierte Personen Zugriff haben.

- Jede Datenbankänderung wird dokumentiert, damit Sie einfach nachvollziehen können, wer welche Änderung zu welchem Zeitpunkt vorgenommen hat.
- Sie müssen über einen Archivierungsmechanismus verfügen, der es Ihnen erlaubt, schnell Einblick in Ihre historischen Daten erhalten zu können.
- Die Kommunikation zwischen den Systemkomponenten muss bestmöglich verschlüsselt werden, damit niemand einen Einblick in die Ausgabedaten vertraulicher Prozesse wie z. B. der Gehaltsabrechnung erhält. Ohne eine wirkungsvolle Verschlüsselung können sich Hacker leicht Zugang zu diesen Informationen verschaffen, indem sie z. B. Netzwerk-Sniffer verwenden.

SECTION 11

Fazit

Ein Job-Scheduler bzw. eine Lösung für die Workflow-Automatisierung muss die folgenden Leistungsmerkmale umfassen, wenn sie den Herausforderungen eines modernen Unternehmens gewachsen sein soll:

- Effektives Job-Scheduling
- Integrierter Managed File Transfer
- Zentrale und skalierbare Multi-Tenancy-Architektur
- Objektorientierung für schnelle, flexible und wiederverwendbare Workflows
- Native Unterstützung von Applikationen
- Dynamische und ereignisbasierte Automatisierung
- Berücksichtigung bedingter Geschäftsregeln
- Advanced Workload Balancing
- Self-Service-Bereitstellung für eine optimale Unterstützung der Endnutzer
- Sicherstellung einer optimalen Datensicherheit und Compliance innerhalb des gesamten Unternehmens

Um sichergehen zu können, dass Ihr Unternehmen für alle zukünftigen Anforderungen gewappnet ist, sollten Sie für die Auswahl einer Workflow-Lösung mindestens die oben genannten grundlegenden Punkte berücksichtigen.

Weitere Informationen finden Sie bitte unter: ca.com/de/automation

Kontaktieren Sie CA technologies



CA Technologies (NASDAQ: CA) entwickelt Software, die die Transformation von Unternehmen vorantreibt und ihnen ermöglicht, die Chancen der Application Economy zu nutzen. Software ist das Herz eines jeden Unternehmens, in jeder Branche. Von der Planung über die Entwicklung bis zu Management und Sicherheit arbeitet CA weltweit mit Unternehmen zusammen, um unsere Lebensweise zu verändern in Bezug auf unsere Kommunikation und Interaktion und das unterwegs, zuhause und in öffentlichen Clouds sowie in Mainframe-Umgebungen. Erfahren Sie mehr unter: ca.com/de.

Copyright © 2018 CA. Alle Rechte vorbehalten. UNIX ist ein eingetragener Markenname der The Open Group. Alle Markennamen, auf die in diesem Dokument verwiesen wird, sind Eigentum der jeweiligen Unternehmen. Dieses Dokument dient ausschließlich der Information und enthält keinerlei Gewährleistungen oder Zusicherungen. Die beschriebenen Funktionen sind gegebenenfalls kundenspezifisch und die tatsächliche Produkt-Performance kann variieren.

Einige Informationen dieser Publikation basieren auf den Erfahrungen von CA mit dem genannten Softwareprodukt in unterschiedlichen Entwicklungs- und Kundenumgebungen. Die in der Vergangenheit erzielte Performance des Softwareprodukts in diesen Entwicklungs- und Kundenumgebungen stellt keine Gewähr für die zukünftige Performance des Softwareprodukts in identischen, ähnlichen oder abweichenden Umgebungen dar. CA kann nicht garantieren, dass das Softwareprodukt wie in dieser Veröffentlichung beschrieben funktioniert. CA unterstützt das genannte Produkt lediglich in Übereinstimmung mit (i) der Dokumentation und den Spezifikationen, die zusammen mit dem genannten Produkt bereitgestellt werden und (ii) der zum Zeitpunkt der Veröffentlichung dieser Publikation angebotenen Wartungs- und Supportleistungen des genannten Produkts.

Es können keinerlei Ansprüche dahingehend geltend gemacht werden, dass ein Produkt oder Service von CA Technologies von Kunden eingesetzt werden kann, um gesetzliche Auflagen (finanzieller oder anderer Art) zu erfüllen.

CA bietet keine Rechtsberatung an. Weder dieses Dokument noch irgendein hier genanntes Softwareprodukt stellt einen Ersatz für die Einhaltung von gesetzlichen Vorgaben dar (einschließlich, jedoch nicht begrenzt auf Verordnungen, Statute, Anweisungen, Regelungen, Maßnahmen, Beschlüsse, Richtlinien, Normen, Standards, Bestimmungen, Anordnungen, Weisungen etc. (allgemein „Gesetze“)). Bitte wenden Sie sich bezüglich weiterer Informationen zu den in diesem Dokument genannten Gesetzen an einen kompetenten Rechtsberater.