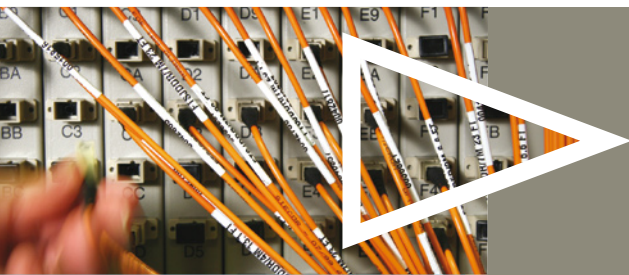




ACTIVATE BUSINESS WITH THE POWER OF IT.™



Was muss eine Configuration Management Database (CMDB) leisten?

Inhalt

EINFÜHRUNG	1
WOZU EINE CMDB?	2
Die Bedeutung des Konfigurations-Managements.....	2
Ziele	2
Vorteile.....	2
Daten sind der Schlüssel.....	3
Entwicklung der CMDB.....	3
EMPFOHLENE ARBEITSWEISE MIT EINER CMDB	6
Inhalt einer ITIL CMDB.....	6
Wie die Stücke zusammenpassen	8
WAS EINE CMDB LEISTEN SOLLTE	10
Föderierung von Daten	10
Flexibles Datenmodell	10
Partitionierung von Konfigurationen.....	11
Abstimmung von Konfigurationen.....	11
Offener Datenzugang	12
ZUSAMMENFASSUNG	13
GLOSSAR	14

Einführung

Vielleicht hat Sie schon einmal jemand daraufhin angesprochen, dass ein Unternehmen eine CMDB (Configuration Management Database) braucht, aber es ist Ihnen vielleicht nicht klar, wozu, und vor allen Dingen, wie man zu einer solchen Datenbank kommt. Dieses Grundsatzdokument von BMC Software zeigt auf der Grundlage der Zielsetzungen der IT Infrastructure Library (ITIL®) die Notwendigkeit eines Konfigurations-Managements und insbesondere einer CMDB auf.

Es werden die hierin unterschiedlichen Datentypen beschrieben, die üblicherweise in einer CMDB gespeichert werden, von CIs (Configuration Item) und ihren Beziehungen bis hin zu verwandten Daten wie zum Beispiel Change Requests und einem Service Impact-Modell. Nach einem kurzen Blick auf die CMDB-Ansätze, die bislang praktiziert wurden, wie zum Beispiel die Integration von Datenspeichern oder die Verwendung einer einzigen zentralen Datenbank wird in diesem Dokument unsere Konzeption einer CMDB vorgestellt: Ein föderiertes Modell, in dem sich Ihre Datenquellen für CIs befinden, bei dem aber Ihre Investitionen in vorhandene Datenbanken erhalten bleiben.

Es werden die Merkmale dargestellt, die eine CMDB haben muss, wie zum Beispiel ein flexibles Datenmodell, die Partitionierung von Konfigurationen, der Abgleich von Konfigurationen und ein offener Datenzugang. Das Dokument erklärt, in welcher Weise diese Merkmale jeweils einen Beitrag zu einer umfassenden Konfigurations-Management-Lösung liefern, die Ihrer IT-Abteilung bestmöglichen Service ermöglicht.

Wozu eine CMDB?

Nachdem in IT-Abteilungen ein wachsendes Interesse daran besteht, Best Practices einzuführen, insbesondere nach Standards wie zum Beispiel ITIL, entscheiden sich heute immer mehr Unternehmen für die Einführung einer Configuration Management Database (CMDB). Sie haben erkannt, dass es sich für ihr Unternehmen lohnt, über eine einzige Quelle zu verfügen, die ein logisches Modell der IT-Infrastruktur liefert und mit der sich alle Configuration Items (CI) in der Umgebung identifizieren, verwalten und verifizieren lassen.

Die Bedeutung des Konfigurations-Managements

IT-Abteilungen sehen sich zahlreichen Herausforderungen gegenüber, wenn sie in zuverlässiger Weise arbeiten und damit die geschäftlichen Zielsetzungen ihres Unternehmens unterstützen wollen. Um diese zu bewältigen, ist vor allen Dingen eine gute Konfigurations-Managementstrategie erforderlich: Wenn man nicht weiß, was sich in der Umgebung befindet, kann man es auch nicht überwachen, pflegen oder verbessern.

Ziele

Gemäß dem ITIL-Handbuch „Service Support“ sollte das Konfigurations-Management die folgenden Ziele haben:

- > Erfassung aller IT-Vermögenswerte und -Konfigurationen innerhalb der Organisation und ihrer Dienste
- > Zurverfügungstellung genauer Informationen über Konfigurationen und deren Dokumentation zur Unterstützung der übrigen Service Management-Prozesse
- > Schaffung einer tragfähigen Basis für Incident Management, Problem Management, Change Management und Release Management
- > Überprüfung der Konfigurationsdaten anhand der Infrastruktur und Beseitigung von Abweichungen

Vorteile

Die Verwirklichung dieser Ziele bedeutet für Ihr Unternehmen signifikante, messbare Verbesserungen hinsichtlich Kontrolle, Integration und Entscheidungsunterstützung.

Kontrolle

Mit der Überprüfung und Berichtigung von Konfigurationsdatensätzen erhalten Sie mehr Kontrolle über Ihre Infrastruktur. So verringern Sie zum Beispiel durch Kontrolle der Versionen der Configuration Items die Komplexität Ihrer Umgebung und senken damit die Kosten für den Desktop-Support. Artikel, die ohne Bezahlung verschwinden oder auftauchen, werden erfasst, sodass Sie Kontrolle über Ihre Vermögenswerte haben und juristische Probleme vermeiden. Mehr Kontrolle über Ihre Umgebung bedeutet auch, dass Sie die allgemeine Sicherheit verbessern können.

Integration

Wenn Prozesse wie Incident Management, Problem Management, Change Management und Release Management auf aktuellen Daten Ihrer Konfiguration basieren, können Sie zusammengefasst werden, wodurch Verwaltungskosten eingespart und Fehler vermieden werden können. So könnten Sie zum Beispiel Incident Management- und Change Management-Prozesse auf zweierlei Arten integrieren:

- > Wenn die Behebung eines Vorfalles eine Änderung erforderlich macht, kann die Incident Management-Anwendung eine entsprechende Änderungsanforderung automatisch erzeugen.
- > Eine Incident- oder Problem-Management-Anwendung kann ein Service-Modell zur Identifizierung bisheriger Veränderungen heranziehen, die eine Störung verursacht haben könnten.

Eine Integration aller konfigurationsbezogenen IT-Prozesse kann den Personalbedarf für die Verwaltung Ihrer Umgebung reduzieren und dadurch zu Einsparungen führen.

Entscheidungsunterstützung

Es kommt Ihren IT-Verwaltern zugute, wenn solche genauen Konfigurationsinformationen in Ihre Service Management-Prozesse übernommen werden. Entscheidungen zu treffen ist einfacher, wenn man über vollständige und genaue Daten verfügt, die bessere Abschätzungen hinsichtlich der Ressourcen und der Performance erlauben. Die Zuordnung zu Serviceebenen steht auf einer festeren Basis, und das Risikomanagement verbessert sich, womit die nicht geplanten Ausfallzeiten sinken.

1. Office of Government Commerce, Best Practice for Service Support (London: The Stationery Office, 2000).

Daten sind der Schlüssel

Sie können Ihr Konfigurations-Management mit einem der oben erwähnten Prozesse oder mit verschiedenen anderen beginnen. Aber gleichgültig, welche Konfigurations-Managementprozesse Sie durchführen - sie greifen nur dank der Daten, die sie verwenden.

Konfigurationsdaten müssen genau sein. Das bedeutet, dass sie häufig aktualisiert werden müssen. Konfigurationen verändern sich laufend. Daten, die vergangene Woche noch richtig waren, können in dieser Woche schon völlig überholt sein. Dann kann es geschehen, dass zehn Server gekauft werden, wenn Sie nur fünf brauchen, oder dass sogar ein Sicherheitspatch aufgespielt wird, der zu einem Systemausfall führt.

Ihre Konfigurationsdaten müssen weiterhin für alle IT-Prozesse zur Verfügung stehen, weil auch die genauesten Daten nichts nützen, wenn man keinen Zugang zu ihnen hat. Wenn zum Beispiel die Netzwerktopologie-Daten Ihrer Discovery-Anwendung Ihrer Change Management-Anwendung nicht zur Verfügung stehen, dann ist es unmöglich, eine intelligente Netzwerk-Neuplanung durchzuführen.

Die Lösung, die es Ihnen erlaubt, genaue Konfigurationsdaten zu pflegen, die mehreren IT-Prozessen zur Verfügung stehen, ist eine CMDB.

Entwicklung der CMDB

Die Konzeption einer CMDB hat sich im Laufe der Jahre aus einer Sammlung isolierter Datenspeicher über integrierte Datenspeicher hin zu einer einzelnen, zentralen Datenbank entwickelt, die als Datenquelle für Konfigurationsdaten dienen kann, ohne Ihre Infrastruktur zu belasten.

Isolierte Datenspeicher

Zu Beginn bestand eine CMDB einfach aus verschiedenen Anwendungen, die jeweils ihre eigenen Daten speicherten, und vielfach weiteren Datenbanken mit Konfigurationsdaten (siehe **Abbildung 1**).

Mit diesem Ansatz konnte das erste ITIL-Ziel einer Nachweisführung für IT-Wirtschaftsgüter und -Dienstleistungen erreicht werden, nicht aber die übrigen, weil die Daten nicht integriert waren. Ihre Asset Management-Anwendung konnte keine Daten von einer Discovery-Anwendung erkennen, und Ihr Service Impact Management konnte keine Service Level Agreements (SLAs) ändern.

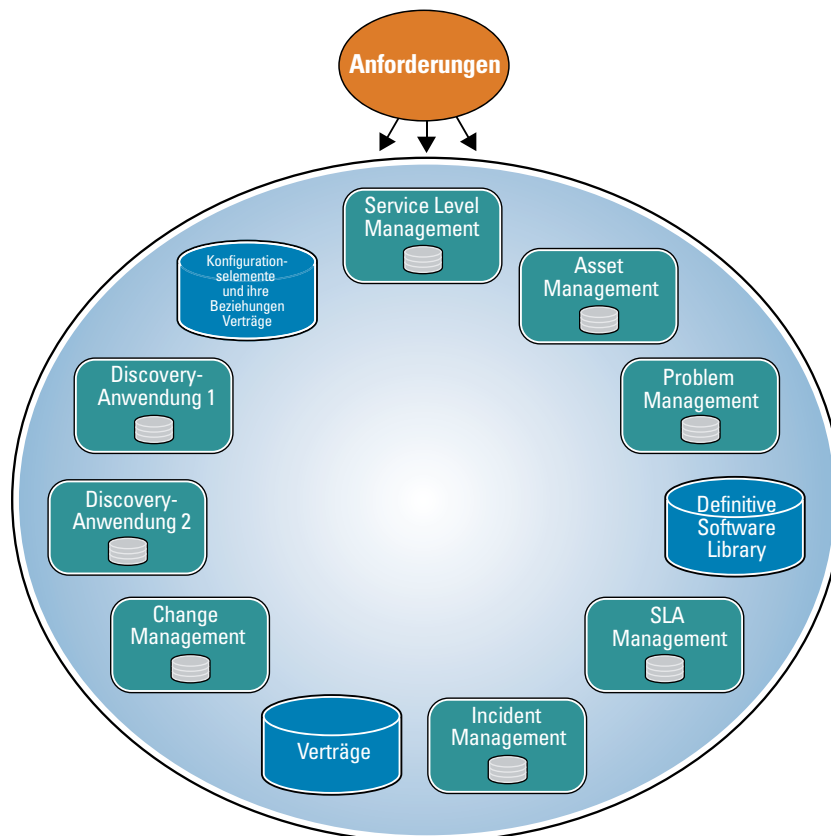


Abbildung 1: Isolierte Datenspeicher

Ein weiterer Nachteil war das Fehlen eines einheitlichen Zugangspunkts, weshalb jeder, der Daten benötigte, wissen musste, wo er sie finden und wie er sie abrufen konnte. Und schließlich war es mit dieser Methode nicht möglich, Informationen über die Beziehungen zwischen CIs zu speichern. Bezüglich weiterer Informationen über Beziehungen siehe „Beziehungen zwischen CIs“ im Abschnitt „Empfohlene Arbeitsweise mit einer CMDB“ dieses Dokuments.

Integrierte Datenspeicher

Als nächstes erzeugten IT-Organisationen CMDBs durch direkte Integration ihrer verschiedenen Datenquellen und Anwendungen, indem sie jeden Datenverbraucher mit jedem Anbieter verbanden, von dem er Daten benötigte (siehe **Abbildung 2**).

Mit dieser Methode konnten unterschiedliche Konfigurations-Managementprozesse Daten gemeinsam nutzen, was den Nutzwert der CMDB erheblich verbesserte. Es waren jedoch erhebliche Ressourcen erforderlich, um die verschiedenen Integrationen herzustellen und zu pflegen. Und wie bei den isolierten Datenspeichern wusste jemand, der mit dem System nicht vertraut war, nicht unbedingt, wo er bestimmte Daten suchen sollte.

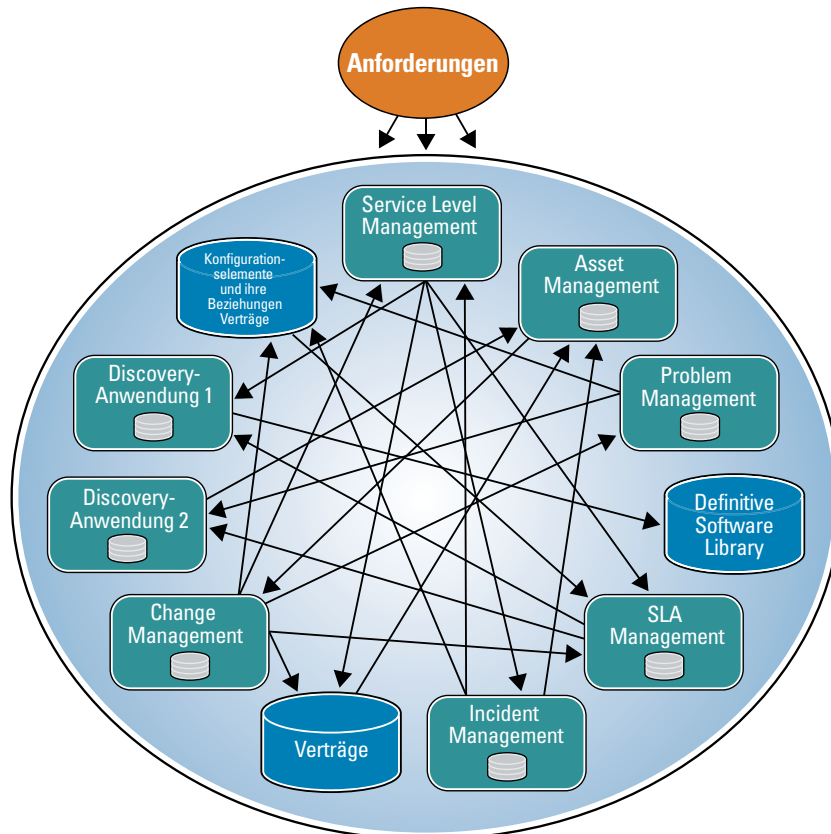


Abbildung 2: Direkte Integration ungleichartiger Datenspeicher

Eine zentrale Datenbank

Seit kurzem werden einzelne, alles umfassende CMDB für Konfigurationsdaten angeboten, auf die alle Anwendungen zugreifen können, die die Daten benötigen (siehe **Abbildung 3**).

Bei dieser Methode können alle Anwendungen, die in die CMDB integriert sind - sowohl Verbraucher als auch Anbieter von Daten - auf alle konfigurationsbezogenen Daten zugreifen, womit die gemeinsame Nutzung wieder einen Schritt über die Methode der integrierten Datenspeicher hinausgeht. Sie bietet weiterhin einen einzigen Zugangspunkt, womit die CMDB zu einer Datenquelle wird, bei der Benutzer alle Anfragen stellen können.

Aber eine alles umfassende Datenbank hat wiederum Nachteile. Sie erfordert eine große Kapazität an einem einzigen Ort und schafft einen Engpass, weil alle Anforderungen von Daten und deren Aktualisierungen über denselben Pfad laufen. Weiterhin ist ein enormer Migrationsaufwand erforderlich, um alle Daten in die einzige Datenbank zu überführen, wodurch ein kompliziertes Datenmodell entsteht, das jedesmal geändert werden muss, wenn sich eine in die CMDB integrierte Anwendung ändert. Und wenn diese Anwendungen nicht vom selben Anbieter wie der CMDB stammen, können solche Integrationen außerordentlich aufwändig sein.

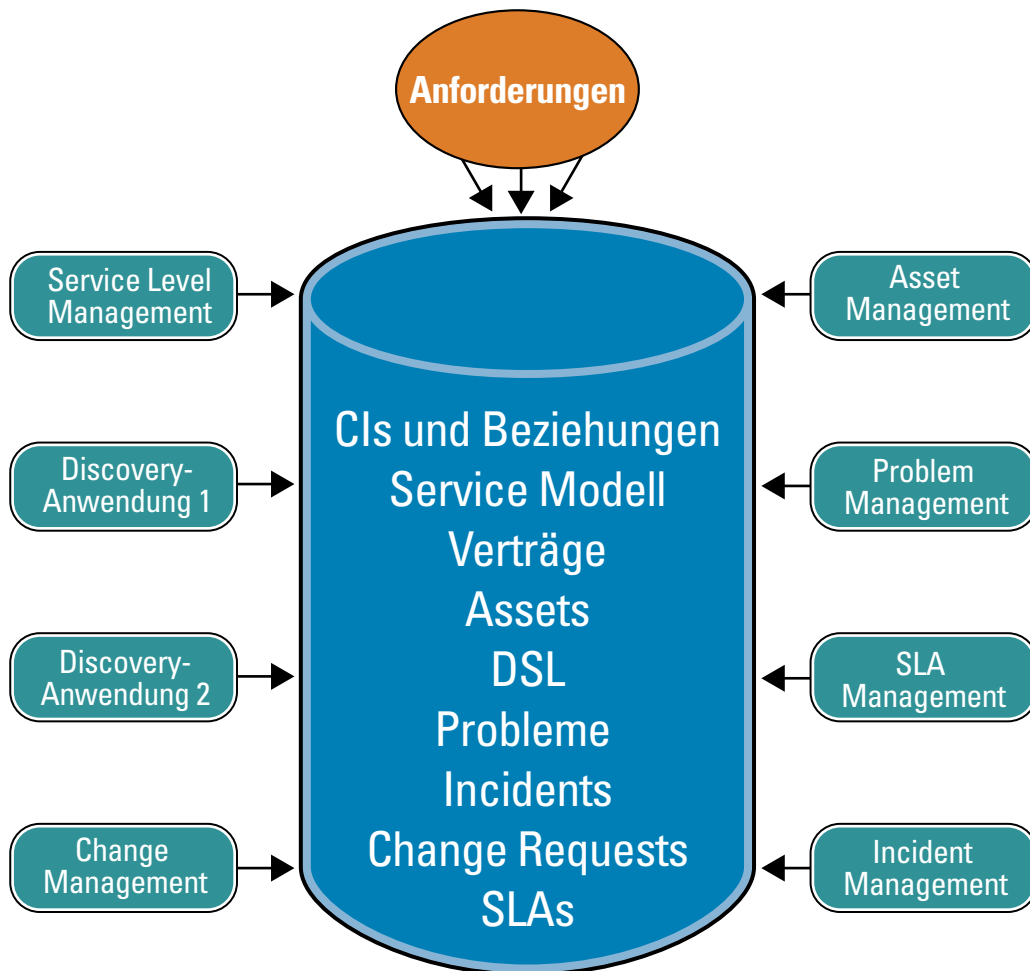


Abbildung 3: Eine einzige zentrale Datenbank

Empfohlene Arbeitsweise mit einer CMDB

BMC Software ist überzeugt, dass eine CMDB mit einem föderierten Datenmodell, das eine zentrale Datenbank mit Anbindung an andere Datenspeicher beinhaltet, die beste Möglichkeit ist, Konfigurationsdaten ohne die hohen Einrichtungs- und Wartungskosten gemeinsam zu nutzen, die der reine zentralisierte Ansatz mit sich bringt. In diesem Abschnitt werden die betreffenden Datentypen beschrieben und anschließend erklärt, wie das föderierte Modell Daten trennt.

Inhalt einer ITIL-CMDB

Gemäß ITIL sollen verschiedene Typen von Daten in der CMDB gespeichert werden. Hauptzweck ist es, CIs und die Beziehungen beisammen zu halten, die miteinander zu einem bestimmten Zeitpunkt oder in einem bestimmten Zustand eine Konfiguration bilden. ITIL empfiehlt weiterhin, dass die CMDB in einem Zusammenhang mit CIs stehende Daten aufnimmt, wie zum Beispiel Help Desk Tickets oder SLA-Definitionen.

Was ist ein Configuration Item?

CIs sind das zentrale Element einer CMDB. Ohne eine klare Definition, was ein CI ist, wird man über Schwierigkeiten haben zu entscheiden, ob bestimmte Arten von Daten in die CMDB gehören oder nicht.

Einfach ausgedrückt, ist ein CI ein Objekt (oder eine Instanz) einer Dateneinheit, die Teil Ihrer Umgebung ist und spezifisch für dieses Objekt konfigurierbare Attribute besitzt. Diese Dateneinheiten können physischer Art (wie z. B. ein Rechnersystem), logischer Art (wie z. B. eine installierte Instanz eines Softwareprogramms) oder konzeptioneller Art sein (wie z. B. ein Geschäftsprozess). Immer aber müssen sie direkter Teil Ihrer Umgebung sein, keine Information über einen solchen Teil. Einige Beispiele sollen helfen, eine Abgrenzung vorzunehmen:

Configuration Items	Keine Configuration Items
Ein Rechnersystem ist Teil Ihrer Umgebung und besitzt konfigurierbare Attribute wie z. B. eine Seriennummer, eine Prozessorgeschwindigkeit und eine IP-Adresse.	Ein Helpdesk Ticket hat konfigurierbare Attribute, ist aber nicht direkter Teil Ihrer Umgebung. Es enthält aber Informationen über andere Dateneinheiten (z.B. ein Rechnersystem), die Teil Ihrer Umgebung sind.
Ein Gebäude ist Teil Ihrer Umgebung und besitzt konfigurierbare Attribute wie z. B. eine Anzahl von Räumen, eine Klimaanlage und eine Alarmanlage.	Ein archiviertes Softwarepaket ist Teil Ihrer Umgebung und wird üblicherweise in der Definitive Software Library (DSL) gespeichert.
Ein Mitarbeiter ist Teil Ihrer Umgebung und besitzt konfigurierbare Attribute wie z. B. Kenntnisse, Arbeitsstunden und eine Abteilung.	Ein Service Level-Vertrag besitzt konfigurierbare Attribute, aber ist kein direkter Teil Ihrer Umgebung. Er enthält Informationen über andere Dateneinheiten (z.B. einen Webserver), die Teil Ihrer Umgebung sind.
Eine auf einem Rechnersystem installierte Softwareinstanz ist Teil Ihrer Umgebung und besitzt konfigurierbare Attribute wie z. B. eine Seriennummer, eine Patch-Ebene und eine Einsatzmethode.	Ein Vertrag hat konfigurierbare Attribute, ist aber nicht direkter Teil Ihrer Umgebung. Er enthält Informationen über andere Einheiten (z.B. einen Kopierer), die Teil Ihrer Umgebung sind.
Ein Geschäftsprozess ist Teil Ihrer Umgebung und besitzt konfigurierbare Attribute wie z. B. die Bedeutung für das Unternehmen und die Kosten einer Unterbrechung des Dienstes.	Ein Ereignis hat keine konfigurierbaren Attribute und ist nicht Teil Ihrer Umgebung.

Tabelle 1: Was sind Configuration Items?

Natürlich muss nicht alles, was die Kriterien eines CI erfüllt, verfolgt werden. So wird man vermutlich in der CMDB keine Datensätze für alle Bürostühle im Unternehmen anlegen wollen.

Abhängigkeiten zwischen CIs

CIs existieren nicht im luftleeren Raum; sie beeinflussen einander. Ein CI kann ein anderes CI verwenden, davon abhängig sein, ein Teil davon sein, es auslösen oder in diesem enthalten sein, um nur einige Beispiele zu nennen.

Speichert man diese Abhängigkeiten in der CMDB, kann man sehen, wie CIs miteinander verbunden sind und sich gegenseitig beeinflussen. Abhängigkeiten können einfach sein, wie z. B. ein Laufwerk ein Element eines Computersystems ist, oder komplexer (siehe **Abbildung 4**).

Abhängigkeiten bestehen nicht nur zwischen physischen CIs, sondern auch zwischen logischen und begrifflichen CIs, wie z. B. die Softwareinstanzen und der Service in **Abbildung 4**. Zwei CIs können mehr als eine Abhängigkeit voneinander haben: So kann z. B. ein Mitarbeiter einen Server besitzen und diesen auch bedienen.

Abhängigkeitsdaten machen die CMDB zu einem relevanten Werkzeug für die Entscheidungsfindung. Dank der Abhängigkeiten und der sonstigen Beziehungen zwischen Ihren CIs können Sie z. B. entscheiden, wie ein Upgrade von Prozessor A die Leistung von Server B verbessern würde, oder welche Dienste betroffen wären, wenn Router C ausfallen würde. Die meisten Stillstandszeiten werden durch Probleme verursacht, die auf Konfigurationsänderungen zurückgehen, und dies sind genau die Informationen, die Ihnen helfen können, dies zu vermeiden.

Verwandte Daten

Mit CIs sind eine Fülle von Informationen verbunden wie zum Beispiel Help Desk Tickets, Change-Ereignisse, Verträge, Service Level Agreements (SLAs), eine Definitive Software Library (DSL) und vieles mehr. Diese Dinge sind zwar selbst keine CIs, enthalten aber Informationen über Ihre CIs und bilden einen wichtigen Teil ihrer IT-Infrastruktur.

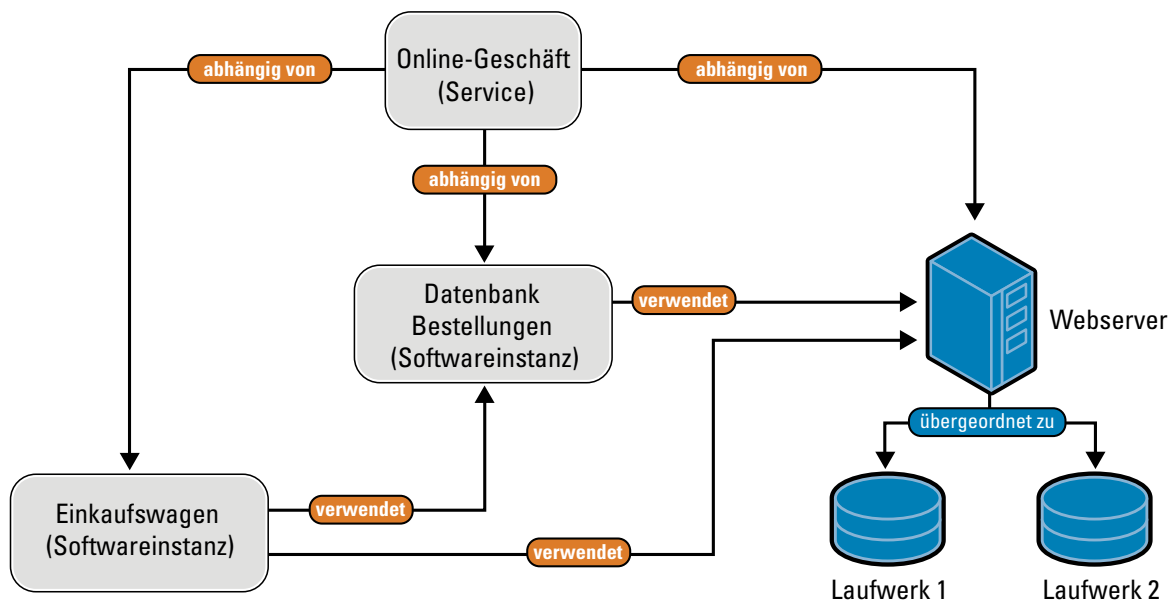


Abbildung 4: Beispiel für Abhängigkeiten

Wie die Stücke zusammenpassen

Die CMDB und ihre Infrastruktur sollten in drei Ebenen untergliedert werden. Dies sind die CMDB selbst, von der CMDB abhängige Daten, die so genannten erweiterten CMDB-Daten, und Anwendungen, die mit diesen beiden Schichten interagieren, die so genannte CMDB-Umgebung (siehe **Abbildung 5**).

Die CMDB und die erweiterten CMDB-Datenebenen bilden zusammen das, was ITIL als CMDB bezeichnet. Diese Auftrennung in zwei Ebenen macht den grundlegenden Unterschied zwischen dem föderierten Ansatz und demjenigen aus, was im Abschnitt „Wozu eine CMDB?“ unter dem Stichwort „Entwicklung der CMDB“ gesagt wurde.

Die CMDB

Die CMDB enthält nur CIs und deren Abhängigkeiten, jedoch können einige ihrer Attribute mit den erweiterten CMDB-Datenebenen verknüpft sein. Es müssen nicht alle verfügbaren CI-Attribute in der CMDB gespeichert werden; grundsätzlich sollten hier nur die Schlüsselattribute gespeichert und Verknüpfungen zu den weniger wichtigen in den erweiterten CMDB-Daten angelegt werden.

Auch wenn die CMDB also nicht alle Attributdaten oder abhängigen Daten enthält, dient sie trotzdem als Datenquelle für die Konfiguration, weil sie Links zu den erweiterten CMDB-Daten enthält. Sie können alle Abfragen an die CMDB schicken, und wenn sich die gesuchten Daten nicht dort befinden, gibt es Verweise auf den Speicherort der Daten und Informationen dazu, wie man Zugriff auf diese Daten erhält.

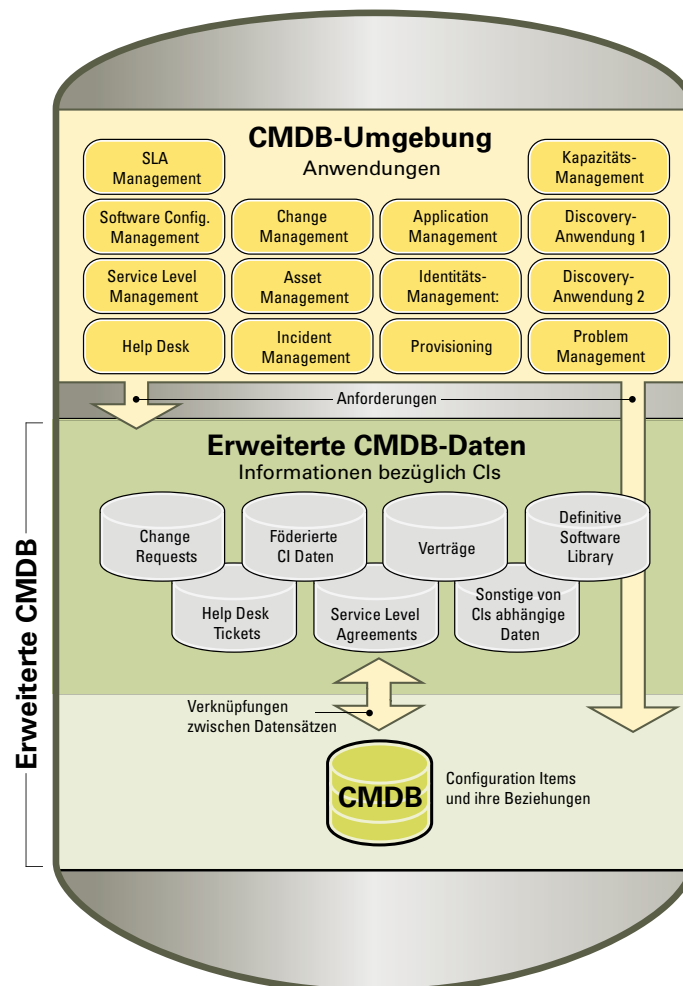


Abbildung 5: Empfohlene CMDB-Infrastruktur mit föderiertem Datenmodell

Die erweiterten CMDB-Daten

Die erweiterten CMDB-Daten sind die im Abschnitt „Empfohlene Arbeitsweise mit einer CMDB“ unter „Verwandte Daten“ genannten Daten sowie sonstige CI-Attribute, für die eine Speicherung in der CMDB als nicht erforderlich gilt.

Die Daten in der Schicht „Erweiterte CMDB-Daten“ sind mit den CI-Daten in der Definition CMDB.By verknüpft; auf föderierte CI-Attribute werden Links von ihren Objekten in der CMDB gelegt, sodass bei Abfragen in der CMDB diese Attribute erreicht werden können. Bei den übrigen Typen erweiterter Daten kann die Verknüpfung in eine oder beide Richtungen gehen. So kann zum Beispiel bei einem Change Request eine Verknüpfung vorhanden sein, bei der man Zugriff auf die Objekte der CIs hat, die dieser ändern wird, und jedes CI-Objekt kann mit einem Link versehen sein, über den man Zugriff auf die Change Requests hat, die sich auf diese auswirken.

Dies hat mehrere Vorteile:

- > Die Funktion der CMDB kann auf CIs und deren Abhängigkeiten beschränkt bleiben. Diese Funktion umfasst Partitionen für mehrere „Schnappschuss“-Versionen, Abgleich von Daten aus mehreren Quellen und föderierte Daten. Näheres hierzu findet sich im Abschnitt „Was eine CMDB leisten sollte“ dieses Dokuments.
- > Die für diese Leistungsmerkmale aufzuwendenden Mittel werden nicht für Daten vergeudet, für die dies nicht erforderlich ist. So sind zum Beispiel mehrere Schnappschüsse einer DSL unnötig; es wäre also eine Vergeudung von wertvollem Speicherplatz, die DSL in die CMDB aufzunehmen.
- > Sie brauchen die CMDB nicht zu ändern, um abhängige Daten aufzunehmen. Mit der Abgrenzung, die für CIs und ihre Abhängigkeiten gegeben wurde, ist die Frage bereits beantwortet, ob ein bestimmter neuer Datentyp in der CMDB gespeichert werden soll: Sie speichern ihn unter den erweiterten CMDB-Daten und sparen sich dadurch die Mühe, bei jedem neuen Datentyp das Datenmodell in der CMDB ändern zu müssen.
- > Sie vermeiden auch die Probleme, die mit einer Verkleinerung des Datenmodells verbunden sind, wenn man sich später entschließt, Daten aus der CMDB auszugliedern.
- > Transaktionsdaten können in Datenbanken gespeichert werden, die große Volumina von Anfragen besser bewältigen als die CMDB.
- > Daten werden auf effizientere Weise zur Verfügung gestellt. Statt alle Daten von der CMDB zu beziehen, können Datenverbraucher sie von einzelnen Datenspeichern beziehen, die für die Bereitstellung des jeweils abgefragten Datentyps optimiert sind.

- > Sie brauchen nicht mehrere Datenmigrationen und Anwendungsintegrationen durchzuführen, um Ihre Change Requests, Help Desk Tickets und sonstige CI-relevante Daten in die CMDB zu übernehmen. Anwendungen, die mit diesen Daten arbeiten, können auf diese weiter dort zugreifen, wo Sie sie derzeit speichern.
- > Die CMDB wird nicht zum Engpass. Da Anforderungen von abhängigen Daten als solche von anderen Datenbanken bearbeitet werden, wird die CMDB nicht zusätzlich zu CI-Abfragen mit diesem Aufkommen belastet. Sie können die Belastung auf mehrere Systeme verteilen.

Es ist zwar möglich, die erweiterten CMDB-Daten an einem einzigen Ort zu speichern, aber nicht notwendig. Die verschiedenen Datentypen in dieser Ebene sind nicht notwendigerweise miteinander verknüpft oder voneinander abhängig. Das einzige, was ihnen gemeinsam sein muss, ist eine Verknüpfung von der oder zur CMDB

Die erweiterte CMDB

Die CMDB und die erweiterten CMDB-Daten bilden die erweiterte CMDB. Dies entspricht dem Begriff „CMDB“ im Sinne der ITIL.

Die CMDB-Umgebung

Während die erweiterte CMDB Daten enthält, ist die CMDB-Umgebung für die Anwendungen da, die diese Daten zur Verfügung stellen und verbrauchen. Diese Anwendungen können Zugriff auf die CMDB, die erweiterten CMDB-Daten oder auf beide haben. So ist zum Beispiel eine Asset Management-Anwendung, die CI-Instanzen in der CMDB anzeigt und ändert, Teil der CMDB-Umgebung als Verbraucher, und eine Discovery-Anwendung, die CI-Instanzen in der CMDB erzeugt, ist Teil der CMDB-Umgebung als Anbieter.

Diese Anwendungen speichern manchmal ihre Informationen in ihren eigenen Datenbanken, aber diese Komponenten gelten dennoch als Bestandteil der verschiedenen Ebenen der CMDB-Infrastruktur. Eine Anwendung ist Teil der CMDB-Umgebung, während ihre konfigurationsbezogenen Daten Teil der erweiterten CMDB sind. Natürlich können Anwendungen in der CMDB-Umgebung auch auf Daten zugreifen, die nicht von CIs abhängig sind. Diese Daten sind kein Teil der erweiterten CMDB.

Was eine CMDB leisten sollte

Nach der Lektüre des vorherigen Abschnitts „Empfohlene Arbeitsweise mit CMDB“ wissen Sie jetzt, wie Sie Ihre CMDB strukturieren und in Ihre übrige Infrastruktur einbinden müssen. Neben der richtigen Struktur braucht eine CMDB aber auch noch weitere Merkmale, damit sie Ihre CIs effektiv verwalten kann. Diese sind:

- > Föderierung von Daten
- > Flexibles Datenmodell
- > Partitionierung von Konfigurationen
- > Abstimmung von Konfigurationen
- > Offener Datenzugang

Föderierung von Daten

Da wir dieses Thema in diesem Dokument bereits angesprochen haben, wissen Sie schon, dass Föderierung mit einem zentralen Ablageort zu tun hat, der bestimmte Daten direkt beinhaltet, während er auf andere Daten in anderen Quellen mittels eines Links verweist.

Sie können bestimmte Attribute föderieren, wenn Sie sie verfolgen möchten, aber nicht so oft und intensiv wie die Schlüsselattribute eines CIs. Diese sekundären Attribute sind die ersten von zwei Arten von Daten, die Sie föderieren können.

Das bedeutet zum Beispiel, dass der CMDB-Datensatz für einen Mitarbeiter ein Attribut „Kenntnisse“ beinhaltet könnte, das eine Auflistung der Kenntnisse des Mitarbeiters beinhaltet, und ein Attribut „Abteilung“, das den Namen der Abteilung des Mitarbeiters beinhaltet. Er könnte auch eine Beziehung mit einem HR-Datenspeicher haben, in dem zusätzliche Attribute wie das Gehalt gespeichert werden, die unter dem Blickwinkel der Konfiguration nicht so wichtig sind.

Der andere Typ sind Daten, die eine Beziehung zu CIs haben, aber keine echten Attribute eines CIs sind, d.h. Daten, die auf ein CI verweisen oder auf die ein CI verweist und durch die das CI zusätzlichen Inhalt oder erweiterte Funktionen erhält, ohne dass diese Daten Teil des CI selbst wären.

So könnten zum Beispiel Ihre CI-Datensätze für Softwareinstanzen eine Lizenz-Beziehung haben, die die URL zu einer Intranet-Seite enthält, auf der die Lizenz abgelegt ist, oder jeder CI Datensatz könnte eine Probleme-Beziehung aufweisen, die die erforderlichen Informationen für eine Hilfedatenbank bei allen Problemen mit diesem CI enthält.

Vorzüge föderierter Daten sind unter anderem:

- > Vermeidung der Gemeinkosten für Import, Verfolgung und Abgleich der Daten in der CMDB.
- > Es besteht ein Standardverfahren für die Verknüpfung abhängiger Daten.
- > Die föderierten Daten können an mehreren Standorten liegen.
- > Die Investition in andere Datenspeicher bleibt erhalten.

Flexibles Datenmodell

Es gibt viele verschiedene Arten von CIs, von Rechnersystemen über Netzwerke-Hardware bis zu Software-Servern. Ohne ein Datenmodell, das diese Arten und die Arten von Beziehungen, die zwischen ihnen bestehen können, genau abbildet, könnte Ihre CMDB Attribute speichern, die nicht zu ihren CIs gehören, notwendige Attribute weglassen und es erschweren, Gruppen von CIs zu suchen. Dieses Datenmodell muss objektorientiert und erweiterbar sein.

Objektorientiert

Ein objektorientiertes Datenmodell besitzt eine hierarchische Menge von Klassen, wobei jede Klasse die Attribute ihrer Überklasse - der ihr in der Hierarchie übergeordneten Klasse - erbt und die eigenen Attribute hinzufügt, um einen spezifischeren Objekttyp zu erzeugen, eine Unterklasse. Unterklassen können wiederum Unterklassen haben, so dass die Hierarchie auf jede beliebige Detailebene ausgebaut werden kann.

So könnte zum Beispiel die Klasse Computersysteme die Attribute Domäne, Prozessortyp und Hersteller haben. Die Klasse Computersystem könnte Unterklassen mit den Bezeichnungen Laptop, Desktop und Großrechner besitzen. Alle diese Unterklassen besitzen die drei Attribute ihrer Überklasse sowie eigene spezifische Attribute. **Abbildung 6** zeigt einen Teil eines objektorientierten CMDB-Datenmodells mit einer Überklasse und zwei Ebenen von Unterklassen.

Die Vorzüge eines objektorientierten Datenmodells sind unter anderem, dass gemeinsame Attribute bei ähnlichen Typen von CIs erzwungen werden, und die Möglichkeit, nicht nur innerhalb einer gegebenen Klasse von CIs, sondern in jedem Zweig der Hierarchie zu suchen. Wenn das Datenmodell eine Basisklasse besitzt, der alle anderen hierarchisch untergeordnet sind, kann man alle CIs und deren Abhängigkeiten suchen.

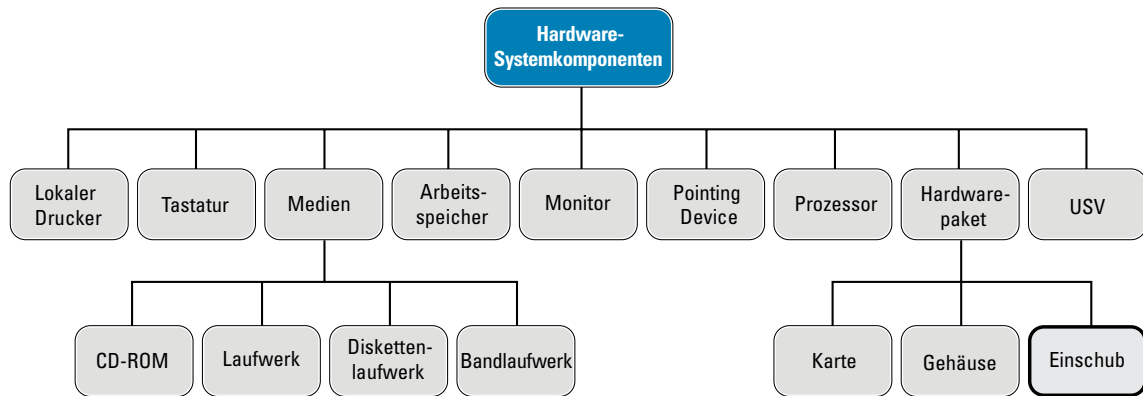


Abbildung 6: Ausschnitt aus einem objektorientierten Datenmodell

Erweiterbar

Ihre Infrastruktur und die darin enthaltene Technik ändern sich ständig. Das bedeutet, dass sich die Arten von CIs und Abhängigkeiten in Ihrer CMDB ebenfalls ändern müssen; Sie brauchen also ein erweiterbares Datenmodell. Sie müssen weiterhin die Möglichkeit haben, Attribute aus Ihren Klassen zu entfernen und sogar Klassen hinzuzufügen und zu entfernen.

Dies ist ein wichtiges Merkmal, aber Sie sollten es nicht überstrapazieren. Die CMDB sollte nur häufige CIs und ihre Abhängigkeiten enthalten. Die Hinzufügung von Klassen und Attributen für unwichtige CIs belastet die CMDB nur unnötig. Weiterhin könnte eine zu große Zahl von Unterklassen zu so eng definierten Klassen führen, dass diese nur noch sehr wenige Mitglieder enthalten. Wägen Sie die Notwendigkeit einer Kategorisierung gegenüber der Notwendigkeit ab, ähnliche CIs miteinander zu speichern.

Partitionierung von Konfigurationen

Partitionierung ist die Möglichkeit, die Konfigurationsdaten in als Datenbestände bezeichnete Teile zu zerlegen, die jeweils eine bestimmte Datenmenge zu einem bestimmten Zeitpunkt repräsentieren. Damit können dieselben CI- und Abhängigkeitsinstanzen in mehr als einem Datenbestand vorhanden sein.

Dies ist wichtig, damit Konfigurationsdatensätze anhand der Infrastruktur überprüft und berichtigt werden können. Sie können einen Datenbestand anlegen, der Ihre beabsichtigte Konfiguration wiedergibt, dann mit einer Discovery-Anwendung einen weiteren Datenbestand anlegen, der Ihre aktuelle Konfiguration wiedergibt, und die beiden miteinander vergleichen.

Partitionierung ist ein mächtiges Werkzeug, das man auch für viele andere Zwecke einsetzen kann. Datenbestände könnten folgendes darstellen:

- > Eine veraltete Konfiguration
- > Eine zukünftige Konfiguration
- > Eine überprüfte Referenz-Konfiguration

- > Verschiedene Versionen einer aktuellen Konfiguration
- > Teilmengen einer allgemeinen Konfiguration
- > Daten von verschiedenen Discovery-Anwendungen
- > Daten von verschiedenen Client-Konfigurationen (Multi-Tenancy)
- > Sonstiges nach Ihrer Wahl

Abstimmung von Konfigurationen

Wenn mehrere Datenbestände dieselben Instanzen beinhalten, werden mittels einer Abstimmung die verschiedenen übereinstimmenden Instanzen in allen Datenbeständen ermittelt und dann entweder die verschiedenen Versionen einer jeden Instanz miteinander verglichen und die Unterschiede festgehalten oder die Datenbestände zu einem neuen Datenbestand zusammengefasst. Damit lassen sich Veränderungen im Laufe der Zeit festhalten oder eine gewünschte Konfiguration ermitteln, wenn man über Daten aus mehreren Discovery-Quellen verfügt.

Identifizierung von Instanzen

Bevor man verschiedene Versionen von etwas vergleichen kann, muss man zunächst feststellen, ob sie tatsächlich dieselbe Entität darstellen. Dies wird mittels einer Identifizierung erreicht, wobei Regeln angewandt werden, die man anhand von Instanzen derselben Klasse in zwei oder mehr verschiedenen Datenbeständen festlegt.

So kann z. B. eine Regel, mit der Rechnersystem-Instanzen identifiziert werden sollen, lauten, dass die IP-Adressen beider Instanzen gleich sein müssen. Wenn die Regeln eine Übereinstimmung feststellen, werden beide Instanzen mit derselben Identität gekennzeichnet, wobei ein zusätzliches Attribut angibt, dass beide in ihren jeweiligen Datenbeständen dasselbe Element darstellen.

Vergleich von Datenbeständen

Bei einem Vergleich werden Instanzen in zwei Datenbeständen geprüft und ein Bericht erstellt, der diejenigen Instanzen enthält, die nur in einem der Datenbestände enthalten sind, und die Unterschiede zwischen Instanzen angibt, die in beiden erscheinen. Bei einem Vergleich dürfen nur Instanzen berücksichtigt werden, denen eine Identität zugewiesen wurde.

Bei einem Vergleich kann eine erwartete Konfiguration mit einer aktuellen verglichen werden, was man zu verschiedenen Zwecken nutzen kann. So könnte man sich mit einem Vergleich darauf aufmerksam machen lassen, dass sich in einer Konfiguration etwas geändert hat, die eigentlich statisch bleiben sollte. Oder man könnte mittels eines Vergleichs überprüfen, ob bei einer Veränderungsanforderung die Konfiguration den gewünschten neuen Zustand erreicht hat.

Zusammenfassung von Datenbeständen

Bei einer Zusammenfassung werden nach von Ihnen angegebenen Regeln zwei oder mehr Datenbestände zu einem neuen, einheitlichen Datenbestand zusammengeführt. Dies geschieht üblicherweise zu dem Zweck, eine gültige Konfiguration festzulegen, wenn verschiedene Discovery-Anwendungen bezüglich derselben Elemente einander überschneidende Daten liefern.

Bei einer Zusammenführung dürfen nur Instanzen berücksichtigt werden, denen eine Identität zugewiesen wurde. Die Vorrangsregeln sollten Gewichtswerte für die gewünschten Klassen und Attribute in den einzelnen Datenbeständen enthalten. In den neuen Datenbestand werden jeweils für die Klasse oder das Attribut die Werte desjenigen Datenbestandes mit dem höchsten Gewicht übernommen.

Eine Zusammenführungsfunktion ist wichtig, wenn man zwei oder mehr Discovery-Anwendungen hat, die dieselben CIs entdecken. Jede Discovery-Anwendung hat im Vergleich mit anderen Stärken und Schwächen, so dass man Vorrangsregeln für die Stärken bilden kann. Damit erhält man ein CI-Instanz mit den besten aller entdeckten Daten.

Offenen Zugang zu Daten

Wie schon gesagt, nützen auch die genauesten Daten nichts, wenn man keinen Zugang zu ihnen hat. Man muss immer beachten, dass man Benutzern und Anwendungen erlauben muss, die CMDB zu lesen und zu beschreiben. Verbraucher betrachten und ändern vorhandene Daten, während Anbieter die Daten erzeugen und ändern. Damit sind mindestens die folgenden Merkmale erforderlich:

- > Programmzugriff: Die CMDB muss eine Schnittstelle zur Anwendungsprogrammierung (API) oder ein anderes Verfahren anbieten, damit Programme die Daten anzeigen und ändern können. Hierzu zählen Instanzdaten und die Klassen des Datenmodells.
- > Laden von Massendaten: Die CMDB muss eine Möglichkeit bieten, mehrere Instanzen gleichzeitig zu importieren, damit Discovery-Anwendungen und andere die Datenbank schnell füllen können.
- > Datenbank- und Plattform-Unabhängigkeit: Die CMDB sollte mit verschiedenen Betriebssystemen und Datenbankanbietern kompatibel sein, damit Sie in Ihrer Umgebung flexibel sind.

Zusammenfassung

Damit die ITIL-Ziele für das Konfigurationsmanagement optimal verwirklicht werden, muss eine CMDB:

- > Nur CIs und ihre Abhängigkeiten speichern, wobei abhängige Daten bei den erweiterten CMDB-Daten gespeichert werden
- > Daten föderieren, so dass die CMDB die Quelle der Daten ist und eine Verknüpfung zu abhängigen Daten und weniger wichtigen Attributen herstellen kann
- > Ein objektorientiertes und erweiterbares Datenmodell unterstützen
- > Die Partitionierung von Konfigurationen unterstützen
- > Die Abstimmung von Konfigurationen unterstützen
- > Einen offenen Zugang zu Daten erlauben

BMC Software bietet eine Produktlinie an, die diesen Vorgaben folgt. Um zu erfahren, wie Ihnen diese Produkte helfen können, Ihre Konfigurations-Managementziele zu erreichen, wenden Sie sich bitte an einen Vertriebsbeauftragten von BMC Software oder besuchen Sie www.bmc.com/cmdb Zur weiteren Vertiefung des Themas Konfigurations-Management und CMDB empfehlen wir die nachfolgenden Literaturquellen:

- > Die ITIL-Handbücher, insbesondere Service Support (erhältlich bei www.ogc.gov.uk)
- > Das Common Information Model (CIM) der Distributed Management Task Force (DMTF), erhältlich unter www.dmtf.org

Glossar

Business Service

Ein Service, den das Unternehmen einem anderen Unternehmen anbietet oder den eine Abteilung für andere Abteilungen im gleichen Unternehmen erbringt. Ein Business Service zeichnet sich dadurch aus, dass er direkt für den Geschäftsbetrieb bzw. die angebotenen Produkte und Leistungen (z. B. Kundensupport, Auftragsabwicklung, Lohnbuchhaltung als Dienstleistung) erforderlich ist. Er hat daher in der Regel einen messbaren Nutzen für seine Abnehmer.

CMDB

Ein Datenspeicher, der alle Configuration Items (CIs) der Unternehmensumgebung enthält. Eine CMDB liefert Informationen als Entscheidungsgrundlage für ITIL-Prozesse wie Incident-, Problem- oder Change-Management.

CMDB-Umgebung

Die Verbraucher- und Anbieter-Anwendungen, die mit Daten in der CMDB und den erweiterten CMDB-Daten arbeiten.

Erweiterte CMDB-Daten

Alle Datenspeicher, gleichgültig, ob voneinander abhängig oder nicht, die föderierte CI-Attribute und CI-abhängige Daten enthalten wie zum Beispiel Help Desk Tickets.

CI

Siehe **Configuration Item (CI)**.

CIM

Das Common Information Model, ein Standard-Datenmodell der Distributed Management Task Force (DMTF).

Vergleich

Feststellung der Unterschiede zwischen zwei Datenbeständen oder Teilen von Datenbeständen.

Configuration Item (CI)

Jede Dateneinheit, die für Betrieb und Bereitstellung eines Service erforderlich ist. Ein CI muss sich eindeutig identifizieren lassen, zumindest einige veränderliche Eigenschaften haben, sich verwalten lassen, in einer CMDB aufgezeichnet sein und über bestimmte Attribute (Name, Kategorie, Beziehungen mit anderen CIs usw.) verfügen.

Verbraucher

Eine Anwendung, die Daten anzeigt oder ändert, die sich bereits in der CMDB befinden. Siehe auch **Anbieter**.

Datenbestand

Eine Gruppe von CI- und Abhängigkeitsinstanzen, die eine Menge von Daten zu einem bestimmten Zeitpunkt repräsentiert. Unterschiedliche Datenbestände können Versionen derselben Instanz beinhalten.

Definitive Software Library (DSL)

Ein Datenarchiv, das eine Kopie der genehmigten Softwareversionen beinhaltet, die von einem Unternehmen verwendet werden. Dient zur Kontrolle installierter Versionen.

Discovery

Ein manuell oder automatisch durchgeführter Prozess, bei dem CIs ermittelt, aufgezeichnet, in einer CMDB abgelegt und später aktualisiert werden.

DMTF

Die Distributed Management Task Force, eine IT-Normierungsorganisation, von der das Common Information Model (CIM) stammt.

DSL

Siehe **Definitive Software Library (DSL)**.

Erweiterte CMDB

Die Kombination von CMDB und erweiterten CMDB-Daten. Siehe auch **(CMDB, Erweiterte CMDB-Daten, CMDB-Umgebung)**.

Zusammenführen

Erzeugung eines neuen Datenbestandes durch Heranziehung von Objektattributen aus zwei oder mehr Datenbeständen nach bestimmten Vorrangsregeln

Föderierte Daten

Daten, die nicht direkt in einem zentralen Datenarchiv wie zum Beispiel einer CMDB gespeichert werden, sondern auf die von diesem Datenarchiv aus verwiesen wird, so dass dieses mehr Daten zur Verfügung stellen kann.

Kennung

Eine eindeutige Zeichenfolge zur Identifizierung einer Instanz, damit diese mit Instanzen in anderen Datenbeständen abgestimmt werden kann, so dass sie später verglichen oder zusammengeführt werden können.

Objekt

Ein einzelnes vorhandenes Exemplar einer bestimmten Klasse.

ITIL

Information Technology Infrastructure Library, ein Bestand an Best Practices, der von der britischen Regierung für die Verwaltung von IT-Dienstleistungen erarbeitet wurde.

Glossar (Fortsetzung)

Zusammenführen

Erzeugung eines neuen Datenbestandes durch Heranziehung von Objektattributen aus zwei oder mehr Datenbeständen nach bestimmten Vorrangsregeln.

Partitionierung

Aufteilung von Daten in Datenbestände.

Anbieter

Eine Anwendung, üblicherweise eine Entdeckungsanwendung, die die CMDB mit Daten füllt. Siehe auch **Verbraucher**.

Abstimmen

Eine bestimmte Kombination von Identifizierung, Vergleich und Zusammenführung von Daten.

Abhängigkeit

Ein Zusammenhang zwischen zwei CIs, die einander in bestimmter Weise beeinflussen.

Datenquelle

Eine endgültige Quelle für einen bestimmten Datentyp. Enthält entweder die Daten oder verweist auf diese.

Unterklasse

Eine objektorientierte Klasse, die von einer anderen Klasse abgeleitet ist, ihrer Überklasse. Die Unterklasse erbt alle Attribute der Überklasse.

Überklasse

Eine objektorientierte Klasse, aus der sich alle Unterklassen ableiten.



ACTIVATE BUSINESS WITH THE POWER OF IT.™

Über BMC Software

BMC Software hilft IT-Organisationen, durch besseres Technologiemanagement echten Mehrwert für das Unternehmen zu schaffen. Unsere branchenführenden Business Service Management-Lösungen stellen sicher, dass die Arbeit der IT-Abteilung je nach ihren geschäftlichen Auswirkungen priorisiert wird. So kann die IT proaktiv zur Erfüllung von Unternehmenszielen wie niedrigeren Kosten, höheren Umsätzen und weniger Risiken beitragen. Auf der Basis gemeinsamer BMC Atrium™ Kerntechnologien helfen BMC Lösungen, komplexe und heterogene IT-Systeme und-Prozesse im Griff zu behalten – egal ob Mainframe oder verteilte Umgebung, Datenbanken oder Anwendungen, Services oder Sicherheit. 1980 gegründet, hat BMC Software inzwischen Niederlassungen auf der ganzen Welt und erzielte im Geschäftsjahr 2006 einen Umsatz von über 1,49 Mrd. US\$. BMC Software. Activate your business with the power of IT. Weitere Informationen unter www.bmc.com

