

11.3 CMDB und Configuration Management System (CMS)

“ Kurz erklärt

Damit Configuration Items und ihre Beziehungen verwaltet werden können, benötigen Organisationen geeignete Werkzeuge.

Die bekannteste Datenbank hierfür ist die **Configuration Management Database (CMDB)**.

ITIL betrachtet jedoch nicht nur eine einzelne Datenbank, sondern ein **Configuration Management System (CMS)**, das Informationen aus verschiedenen Quellen zusammenführt.

Ziel ist es, jederzeit aktuelle und verlässliche Informationen über Services und deren Bestandteile bereitzustellen.

Was ist eine CMDB?

Die Configuration Management Database (CMDB) ist eine Datenbank zur Speicherung von Informationen über Configuration Items und deren Beziehungen.

Eine CMDB enthält beispielsweise Informationen über:

- Server
- Clients
- Netzwerkgeräte
- Anwendungen
- Datenbanken
- virtuelle Maschinen
- Container
- Cloud-Ressourcen
- Services
- Dokumentationen
- Verträge

Die CMDB bildet die Grundlage vieler ITSM-Prozesse.

Aufgaben einer CMDB

Eine CMDB soll:

- Configuration Items speichern,
- Beziehungen dokumentieren,
- Änderungen nachvollziehbar machen,
- Auswirkungen sichtbar machen,
- Informationen zentral bereitstellen,
- Incident Management unterstützen,
- Change Enablement unterstützen,
- Problem Management unterstützen.

Sie dient als zentrale Informationsquelle für viele ITIL Practices.

Was ist ein Configuration Management System (CMS)?

Ein CMS ist mehr als eine einzelne Datenbank.

Es verbindet verschiedene Informationsquellen miteinander.

Ein CMS kann beispielsweise Informationen aus folgenden Systemen zusammenführen:

- CMDB
- Monitoring-System
- Inventarisierung
- Asset Management
- Active Directory
- Cloud-Plattformen
- Virtualisierung
- Ticket-System
- Dokumentationssystem
- Discovery-Werkzeuge

Dadurch entsteht ein umfassender Überblick über die IT-Landschaft.

CMDB und CMS unterscheiden

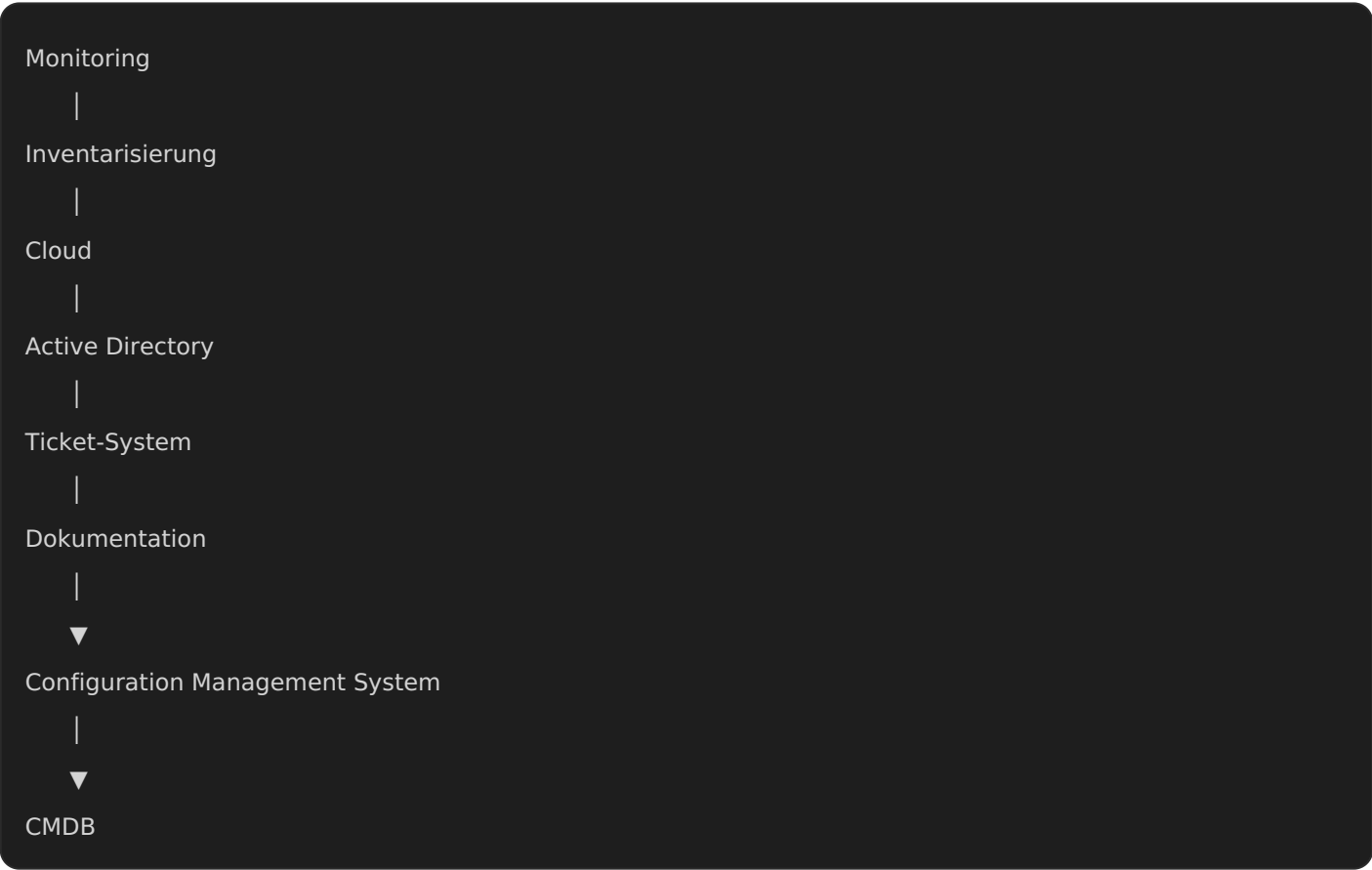
Diese Begriffe werden häufig verwechselt.

CMDB	CMS
einzelne Datenbank	Gesamtsystem
speichert CI-Daten	verbindet mehrere Datenquellen
Bestandteil des CMS	umfasst zahlreiche Systeme
enthält Configuration Items	stellt Informationen bereit

Die CMDB ist also ein Bestandteil des CMS.

Aufbau eines CMS

Ein mögliches Beispiel:



Das CMS stellt Informationen aus verschiedenen Quellen zusammen.

Welche Informationen stammen aus welchen Quellen?

Beispiele:

Informationsquelle	Beispiele
Monitoring	Status, Alarme
Active Directory	Benutzer, Computer
Virtualisierung	virtuelle Maschinen
Cloud	Instanzen, Storage
Inventarisierung	Hardware
Ticket-System	Incidents, Changes
Dokumentation	Runbooks, Architektur

Das CMS verbindet diese Informationen logisch miteinander.

Warum nicht alles in der CMDB speichern?

Nicht jede Information gehört dauerhaft in die CMDB.

Beispiel:

CPU-Auslastung verändert sich ständig.

Diese Information stammt besser direkt aus dem Monitoring.

Die CMDB speichert stattdessen beispielsweise:

- Servername,
- Betriebssystem,
- Verantwortlicher,
- Beziehungen,
- Servicezuordnung.

Aktuelle Zustandsinformationen liefert das Monitoring.

Zentrale Datenhaltung

Eine zentrale Datenbasis bietet zahlreiche Vorteile.

Beispiele:

- einheitliche Informationen,
- weniger doppelte Daten,
- einfachere Pflege,
- bessere Qualität,
- schnellere Suche,
- bessere Auswirkungsanalyse.

Dadurch arbeiten alle ITIL Practices mit denselben Informationen.

Beziehungen visualisieren

Viele CMDB-Systeme können Beziehungen grafisch darstellen.

Beispiel:

CRM-Service

|

└─ Webserver

└─ Datenbank

└─ Storage

└─ Firewall

└─ DNS

Solche Darstellungen erleichtern die Fehlersuche erheblich.

Serviceorientierte Sicht

ITIL betrachtet bevorzugt Services statt Einzelgeräte.

Ein Benutzer interessiert sich beispielsweise dafür,

ob der VPN-Service verfügbar ist,

nicht dafür,

welcher einzelne Server betroffen ist.

Das CMS verbindet beide Sichtweisen.

Abfragen im CMS

Ein CMS beantwortet beispielsweise Fragen wie:

- Welche Server gehören zum ERP-System?
- Welche Datenbank nutzt das Kundenportal?
- Welche Anwendungen laufen auf diesem Server?
- Welche Services sind von diesem Switch abhängig?
- Welche Changes betreffen diesen Service?

Dadurch lassen sich Entscheidungen schneller treffen.

Integration mit Monitoring

Viele CMS-Systeme übernehmen Statusinformationen aus dem Monitoring.

Beispiele:

- Server erreichbar,

- CPU-Auslastung,
- Speicherverbrauch,
- Zertifikatsstatus,
- Netzwerkstatus,
- Verfügbarkeit.

Dadurch stehen aktuelle Informationen direkt zur Verfügung.

Integration mit Change Enablement

Nach einem erfolgreichen Change sollte die CMDB aktualisiert werden.

Beispiele:

- neue Server,
- neue Version,
- neue Beziehungen,
- geänderte IP-Adresse,
- neue Firewallregel,
- neuer Service.

Nur so bleibt die Datenqualität erhalten.

Integration mit Incident Management

Bei einem Incident können Administratoren sofort erkennen:

- betroffene Services,
- abhängige Systeme,
- Verantwortliche,
- Dokumentationen,
- bekannte Beziehungen.

Dadurch verkürzt sich die Analyse erheblich.

Integration mit Problem Management

Problem Management nutzt historische Informationen aus dem CMS.

Beispiele:

- häufig betroffene Systeme,
- wiederkehrende Komponenten,
- Versionen,
- Beziehungen,

- frühere Änderungen.

Diese Informationen unterstützen die Ursachenanalyse.

Discovery-Werkzeuge

Viele Informationen müssen nicht manuell gepflegt werden.

Discovery-Werkzeuge erkennen automatisch:

- neue Server,
- Betriebssysteme,
- installierte Software,
- Netzwerktopologie,
- virtuelle Maschinen,
- Cloud-Ressourcen.

Dadurch sinkt der Pflegeaufwand.

Manuelle Pflege bleibt notwendig

Nicht alle Informationen können automatisch erkannt werden.

Beispiele:

- Servicezuordnung,
- Verantwortlicher,
- Geschäftsprozess,
- Kritikalität,
- Dokumentationen,
- Verträge.

Diese Angaben müssen häufig manuell gepflegt werden.

Datenqualität

Eine CMDB ist nur dann hilfreich,

wenn die gespeicherten Informationen korrekt sind.

Wichtige Kriterien:

- vollständig,
- aktuell,
- eindeutig,

- nachvollziehbar,
- konsistent.

Schlechte Daten führen zu falschen Entscheidungen.

Praxisbeispiel

Ein Administrator plant einen Neustart eines Datenbankservers.

Über die CMDB erkennt er sofort:

Der Server gehört zu:

- ERP,
- Zeiterfassung,
- CRM,
- Reporting.

Dadurch wird der Change besser geplant und betroffene Fachbereiche können rechtzeitig informiert werden.

Typische Fehler

Fehler 1

CMDB und CMS werden gleichgesetzt.

Fehler 2

Configuration Items werden nicht aktualisiert.

Fehler 3

Discovery wird nicht genutzt.

Fehler 4

Monitoring ist nicht integriert.

Fehler 5

Beziehungen fehlen.

Fehler 6

Mehrere Datenquellen liefern widersprüchliche Informationen.

Fehler 7

Zu viele Informationen werden manuell gepflegt.

Fehler 8

Niemand ist für die Datenqualität verantwortlich.

Checkliste CMDB

- ☐ Configuration Items vollständig
 - ☐ Beziehungen dokumentiert
 - ☐ Status aktuell
 - ☐ Verantwortliche gepflegt
 - ☐ Versionen aktuell
 - ☐ Services zugeordnet
 - ☐ Änderungen nachgeführt
 - ☐ regelmäßige Qualitätssicherung
-

Checkliste CMS

- ☐ CMDB integriert
 - ☐ Monitoring angebunden
 - ☐ Discovery aktiv
 - ☐ Active Directory eingebunden
 - ☐ Cloud-Systeme integriert
 - ☐ Dokumentation verknüpft
 - ☐ Ticket-System verbunden
 - ☐ Datenquellen abgestimmt
-

Bedeutung für Fachinformatiker für Systemintegration

Fachinformatiker arbeiten regelmäßig mit CMDBs und CMS-Systemen.

Typische Aufgaben:

- Configuration Items pflegen,
- Beziehungen ergänzen,
- Discovery prüfen,
- Datenqualität verbessern,
- Monitoring anbinden,
- Services dokumentieren,
- Changes nachführen,
- Auswirkungsanalysen durchführen.

Ein gepflegtes CMS erleichtert nahezu alle administrativen Tätigkeiten im IT-Betrieb.

Zusammenfassung

“ Configuration Items erfassen

↓

In der CMDB speichern

↓

Weitere Datenquellen anbinden

↓

Informationen im CMS zusammenführen

↓

Beziehungen darstellen

↓

Informationen für alle ITIL Practices bereitstellen

Merksätze

“ Die CMDB ist ein Bestandteil des CMS.

Ein CMS verbindet Informationen aus mehreren Quellen.

“ Monitoring liefert Zustandsdaten – die CMDB liefert Strukturinformationen.

“ Discovery verbessert die Datenqualität.

“ Eine CMDB ist nur so gut wie die Qualität ihrer Daten.

Verwandte Seiten

- 11.1 Ziele und Grundlagen des Service Configuration Management
- 11.2 Configuration Items, Attribute und Beziehungen
- 11.4 Pflege, Discovery und Datenqualität
- 11.5 Zusammenspiel mit Incident, Problem, Change und Monitoring
- Monitoring and Event Management
- Incident Management
- Change Enablement

Quellen und Versionsstand

Offizielle Grundlagen

- PeopleCert – ITIL Practice Guide: Service Configuration Management
- ITIL Foundation – Version 5

Einordnung

ITIL unterscheidet klar zwischen einer **Configuration Management Database (CMDB)** als Datenbank und einem **Configuration Management System (CMS)** als übergeordnetem Informationssystem. Welche Datenquellen integriert werden und wie die technische Umsetzung erfolgt, hängt von den eingesetzten ITSM-, Discovery- und Monitoring-Lösungen der jeweiligen Organisation ab.

Behandelter Framework-Stand: ITIL Version 5

Zusätzlich berücksichtigt: aktuelle ITIL-4-Practice-Guidance

Fachlicher Stand: August 2026
