

11.2 Configuration Items, Attribute und Beziehungen

“ Kurz erklärt

Ein Configuration Item (CI) besteht nicht nur aus einem Namen.

Erst durch seine Eigenschaften (Attribute) und seine Beziehungen zu anderen Configuration Items entsteht ein vollständiges Bild eines IT-Services.

Ziel ist es, jederzeit nachvollziehen zu können, **welche Komponenten existieren, welche Informationen darüber bekannt sind und wie sie miteinander verbunden sind.**

Aufbau eines Configuration Items

Jedes Configuration Item besitzt eine Reihe von Informationen, die seinen aktuellen Zustand beschreiben.

Typische Bestandteile:

- eindeutige CI-ID
- Name
- Typ
- Beschreibung
- aktueller Status
- Verantwortlicher
- Standort
- Version
- Beziehungen
- Dokumentation

Je kritischer ein CI ist, desto umfangreicher sind häufig die gespeicherten Informationen.

Was sind Attribute?

Attribute sind Eigenschaften eines Configuration Items.

Sie beschreiben das CI genauer.

Beispiele:

- Hersteller
- Modell
- Seriennummer
- Betriebssystem
- IP-Adresse
- MAC-Adresse
- Firmware-Version
- Standort
- Kostenstelle
- Besitzer
- Supportgruppe
- Installationsdatum

Nicht jedes CI benötigt dieselben Attribute.

Beispiel: Server als Configuration Item

Attribut	Beispiel
CI-ID	SRV-00125
Name	APP-SRV-01
Typ	Virtueller Server
Betriebssystem	Windows Server 2025
IP-Adresse	10.20.5.15
Standort	Rechenzentrum Berlin
Status	Produktiv
Verantwortlich	Windows-Team
Service	ERP-System

Diese Informationen erleichtern die Verwaltung erheblich.

Beispiel: Netzwerk-Switch

Ein Switch kann unter anderem folgende Attribute besitzen:

- Hersteller
- Modell
- Seriennummer
- Management-IP
- Firmware-Version
- Rackposition
- VLANs

- Standort
- Verantwortlicher
- Wartungsvertrag

Dadurch lassen sich Wartung und Fehleranalyse vereinfachen.

Pflicht- und optionale Attribute

Nicht alle Informationen sind immer erforderlich.

Beispiel:

Pflichtattribute:

- CI-ID
- Name
- Typ
- Status
- Verantwortlicher

Optionale Attribute:

- Kaufdatum
- Garantie
- Lieferant
- Kosten
- interne Bemerkungen

Die Organisation legt selbst fest, welche Angaben verpflichtend sind.

Eindeutige Identifikation

Jedes Configuration Item sollte eindeutig identifizierbar sein.

Geeignet sind beispielsweise:

- interne CI-ID,
- Asset-Nummer,
- Seriennummer,
- eindeutiger Hostname.

Mehrdeutige Bezeichnungen erschweren den Betrieb.

Beziehungen zwischen Configuration Items

Einzelne Configuration Items bilden selten einen vollständigen Service.

Erst ihre Beziehungen machen Zusammenhänge sichtbar.

Beispiele:

- läuft auf
- verwendet
- verbindet
- gehört zu
- schützt
- repliziert
- sichert
- kommuniziert mit
- ist Bestandteil von

Diese Beziehungen ermöglichen eine deutlich bessere Auswirkungsanalyse.

Beispiel einer Beziehung



Fällt das Storage-System aus,

kann die gesamte Anwendung betroffen sein.

Arten von Beziehungen

Typische Beziehungstypen:

Beziehung	Beispiel
läuft auf	Anwendung → Server
verwendet	Anwendung → Datenbank

Beziehung	Beispiel
verbindet	Switch → Server
schützt	Firewall → Netzwerk
repliziert	Datenbank → Datenbank
enthält	Service → Anwendung
sichert	Backup → Server

Je genauer Beziehungen dokumentiert sind,
desto einfacher lassen sich Auswirkungen erkennen.

Service und Configuration Items

Ein Service besteht meist aus mehreren Configuration Items.

Beispiel:

E-Mail-Service

- |
- ├─ Mailserver
- ├─ Datenbank
- ├─ Storage
- ├─ DNS
- ├─ Firewall
- ├─ Active Directory
- └─ Zertifikate

Der Service ist mehr als die Summe seiner technischen Komponenten.

Auswirkungsanalyse (Impact Analysis)

Beziehungen ermöglichen eine schnelle Bewertung von Auswirkungen.

Beispiel:

Vor einem geplanten Change soll ein Server neu gestartet werden.

Durch die dokumentierten Beziehungen erkennt die IT sofort:

Betroffen sind:

- ERP-System
- Zeiterfassung
- Produktionsplanung

Dadurch kann das Wartungsfenster besser geplant werden.

Root Cause Analysis unterstützen

Auch bei Störungen helfen Beziehungen.

Beispiel:

Monitoring meldet:

- Webserver nicht erreichbar
- API nicht erreichbar
- Kundenportal nicht erreichbar

Die Beziehungen zeigen,

dass alle Systeme dieselbe Datenbank verwenden.

Die eigentliche Ursache liegt vermutlich dort.

Versionen dokumentieren

Viele Configuration Items besitzen unterschiedliche Versionen.

Beispiele:

- Betriebssystem
- Firmware
- BIOS
- Datenbank
- Anwendung
- API
- Container-Image

Versionen erleichtern:

- Fehleranalyse,
 - Sicherheitsupdates,
 - Patchmanagement,
 - Audits.
-

Eigentümer und Verantwortliche

Ein Configuration Item sollte einen fachlichen oder technischen Verantwortlichen besitzen.

Beispiele:

- Windows-Team
- Linux-Team
- Netzwerk-Team
- Datenbankadministration
- Cloud-Team
- Fachbereich

Dadurch ist klar,

wer Änderungen oder Entscheidungen verantwortet.

Status eines Configuration Items

Typische Statuswerte:

- geplant
- Test
- produktiv
- Wartung
- außer Betrieb
- archiviert
- entfernt

Der Status sollte jederzeit aktuell sein.

Dokumentationen verknüpfen

Zu einem CI können verschiedene Dokumente gehören.

Beispiele:

- Installationsanleitung
- Architekturdiagramm
- Runbook
- Wartungsvertrag
- Lizenznachweis
- Sicherheitsrichtlinie
- Notfallhandbuch

Diese Informationen sollten möglichst direkt mit dem CI verknüpft sein.

Automatische Aktualisierung

Viele Attribute können automatisch gepflegt werden.

Beispiele:

- Betriebssystemversion
- IP-Adresse
- Speichergröße
- CPU
- installierte Software
- Firmware-Version
- Seriennummer

Dadurch steigt die Datenqualität.

Praxisbeispiel

Ein Administrator plant,

die Firmware einer Firewall zu aktualisieren.

Durch die Beziehungen erkennt er,

dass betroffen sind:

- VPN
- Internetzugang
- E-Mail
- Webserver
- Cloud-Anbindung

Das Wartungsfenster wird entsprechend geplant.

Typische Fehler

Fehler 1

Configuration Items besitzen keine eindeutige ID.

Fehler 2

Attribute sind unvollständig.

Fehler 3

Beziehungen fehlen.

Fehler 4

Verantwortliche sind nicht dokumentiert.

Fehler 5

Versionen werden nicht aktualisiert.

Fehler 6

Dokumentationen sind nicht verknüpft.

Fehler 7

Status stimmt nicht mit der Realität überein.

Fehler 8

Zu viele irrelevante Attribute werden gepflegt.

Checkliste Configuration Items

- ☐ eindeutige CI-ID vorhanden
 - ☐ Attribute vollständig
 - ☐ Verantwortlicher dokumentiert
 - ☐ Status aktuell
 - ☐ Version gepflegt
 - ☐ Beziehungen vorhanden
 - ☐ Dokumentationen verknüpft
 - ☐ regelmäßige Aktualisierung geplant
-

Checkliste Beziehungen

- ☐ abhängige Systeme dokumentiert

- ☐ Servicezuordnung vorhanden
 - ☐ Netzwerkbeziehungen gepflegt
 - ☐ Datenbanken verknüpft
 - ☐ Cloud-Ressourcen berücksichtigt
 - ☐ Änderungen nachgeführt
 - ☐ Auswirkungen nachvollziehbar
-

Bedeutung für Fachinformatiker für Systemintegration

Fachinformatiker pflegen Configuration Items häufig direkt im Arbeitsalltag.

Typische Aufgaben:

- neue Systeme erfassen,
- Attribute ergänzen,
- Beziehungen dokumentieren,
- Versionen aktualisieren,
- Verantwortlichkeiten pflegen,
- Dokumentationen verknüpfen,
- Auswirkungen von Changes bewerten.

Je vollständiger die Configuration-Daten sind, desto einfacher werden Incident-, Problem- und Change-Prozesse.

Zusammenfassung

“ Configuration Item erfassen



Attribute ergänzen



Beziehungen dokumentieren



Verantwortlichkeiten festlegen



Status aktuell halten



Informationen für andere ITIL Practices bereitstellen

Merksätze

“ Ein Configuration Item besteht aus mehr als seinem Namen.

“ Attribute beschreiben ein CI – Beziehungen verbinden mehrere CIs.

“ Beziehungen sind entscheidend für Impact- und Ursachenanalysen.

“ Aktuelle Attribute erhöhen die Qualität der CMDB.

“ Gute Configuration-Daten erleichtern nahezu alle ITIL Practices.

Verwandte Seiten

- 11.1 Ziele und Grundlagen des Service Configuration Management
- 11.3 CMDB und Configuration Management System (CMS)
- 11.4 Pflege, Discovery und Datenqualität
- 11.5 Zusammenspiel mit Incident, Problem, Change und Monitoring
- Change Enablement
- Incident Management
- Monitoring and Event Management

Quellen und Versionsstand

Offizielle Grundlagen

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

Einordnung

Die dargestellten Attribute, Beziehungstypen und Beispiele orientieren sich an der ITIL Practice „Service Configuration Management“. Welche Attribute gespeichert und welche Beziehungstypen verwendet werden, hängt von den Anforderungen und dem Datenmodell der jeweiligen Organisation ab.

Behandelter Framework-Stand: ITIL Version 5

Zusätzlich berücksichtigt: aktuelle ITIL-4-Practice-Guidance

Fachlicher Stand: August 2026
