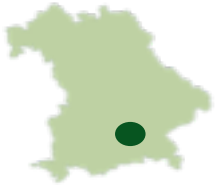
CMDB Patterns

Good Practice für den Entwurf von CMDB-Informationsmodellen

Michael Brenner

Das “Business” des LRZ

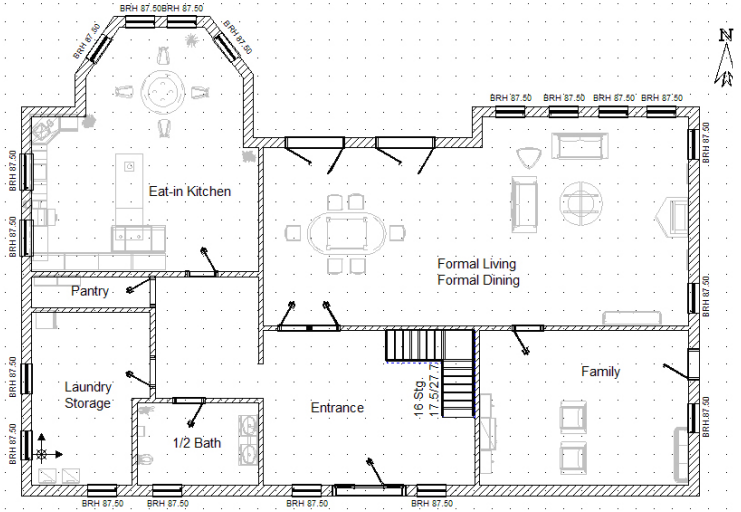


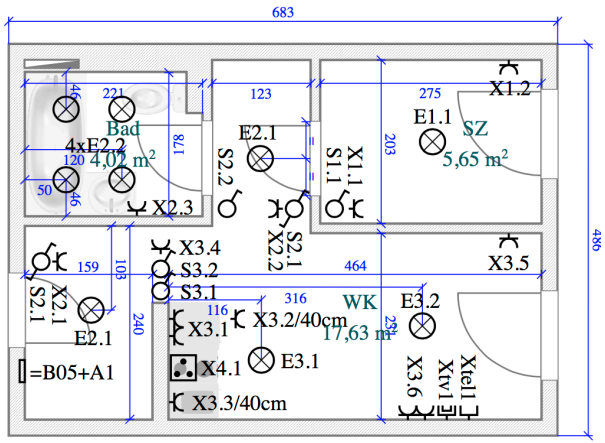
- Nationales und europäisches Höchstleistungsrechenzentrum
  - Gauss Centre for Supercomputing (GCS)
  - HPC- und Grid-Projekte
- Regionales Rechenzentrum für die bayerischen Universitäten
  - Capacity-Computing, Backup und Archivierung
  - Kompetenz-Center (Beratung etc.)
- Rechenzentrum für die Münchner Universitäten
  - Ca. 95.000 Studenten
  - Ca. 30.000 Angestellte

Stark diversifiziertes IT-Service-Angebot, heterogene Infrastruktur

CMDB als Modell

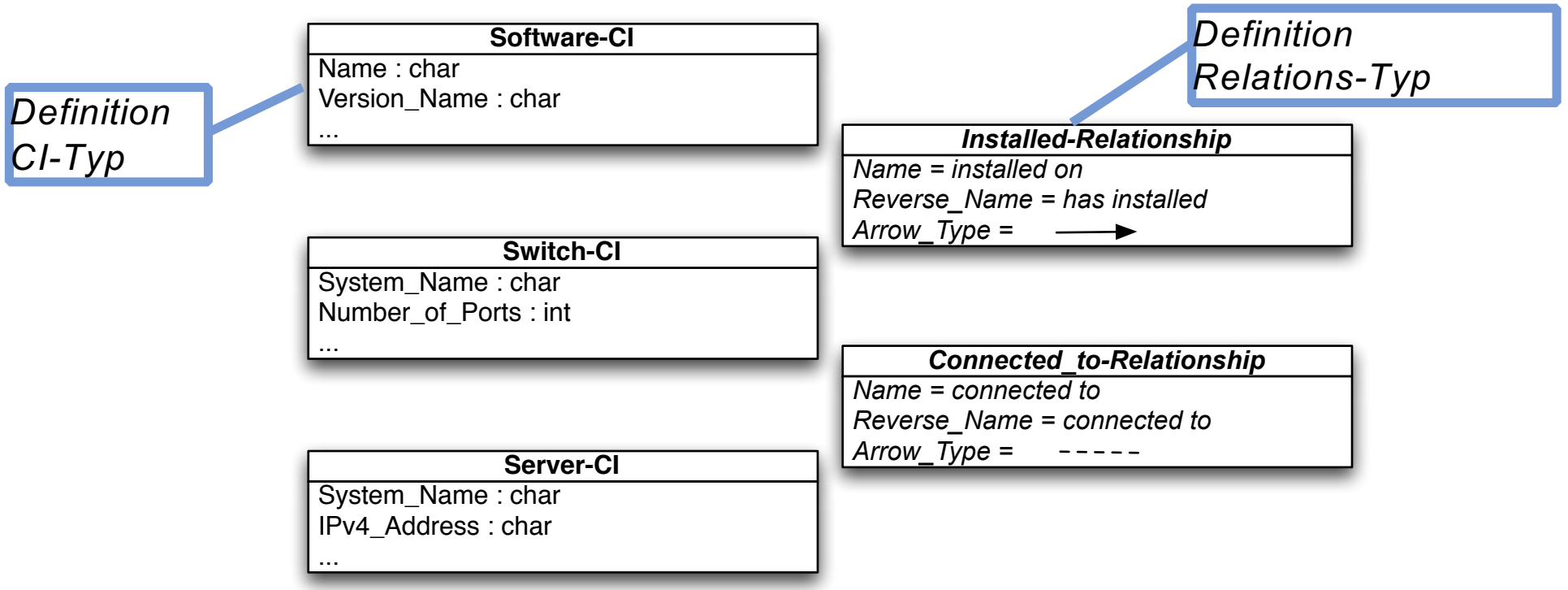
- Modelle sind vereinfachende Abbildungen, die je nach Grad und Art der Vereinfachung besser oder schlechter für bestimmte Anwendungsfälle geeignet sind.
- Metamodelle werden verwendet um die Darstellungssprache von Modellen festzulegen.





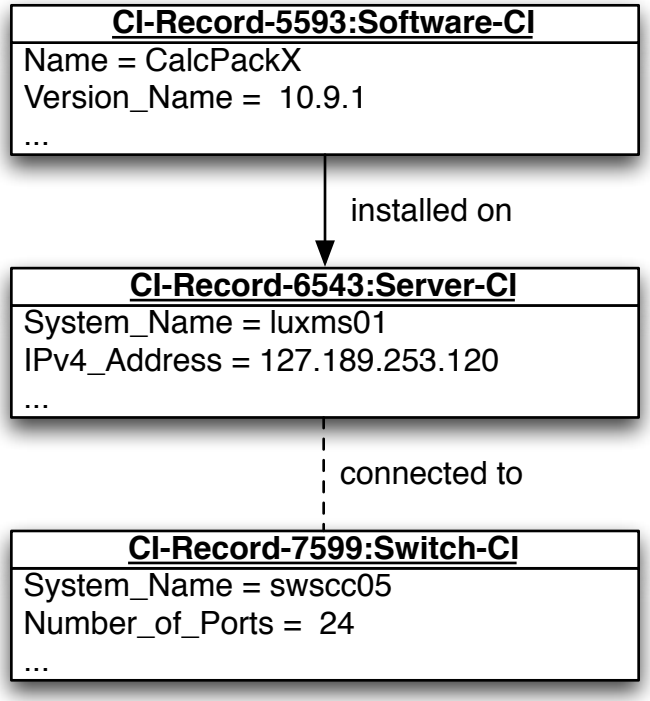
CMDB-Informationsmodell

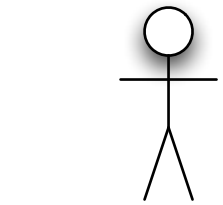
- Datenmodell / Informationsmodell: Bezeichnung für Metamodelle im Netz- und Systemmanagement
- Beispiele: DMTF Common Information Model (CIM), TMForum Shared Information Data Model (SID)
- Typische Form eines out-of-the-box CMDB-Informationsmodells:



Das Configuration-Manager-Dilemma

Instanziierung des
out-of-the-box Datenmodells





Mit dem out-of-the-box Informationsmodell kann ich in unserer komplexen Umgebung n iemals alle Anforderungen erfüllen

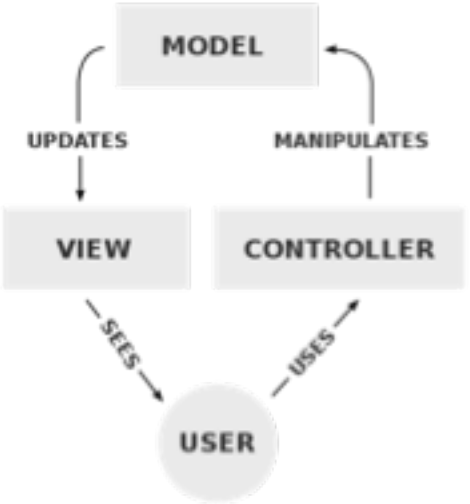
Welche CI- und Relations-Typen soll ich definieren?
Ist eine IP-Adresse ein CI oder ein Attribut? Was ist mit MAC-Adressen?
Wie viele CIs und Relationen gibt das am Ende? Kann man das noch pflegen?

Es gibt so viele Möglichkeiten.
Welche ist die beste?
Gibt es kein ITIL-Buch oder Standard dafür?

Wenn ich das CMDB-Design jetzt falsch mache, gibt das am Ende viel Ärger ...

CMDB Patterns

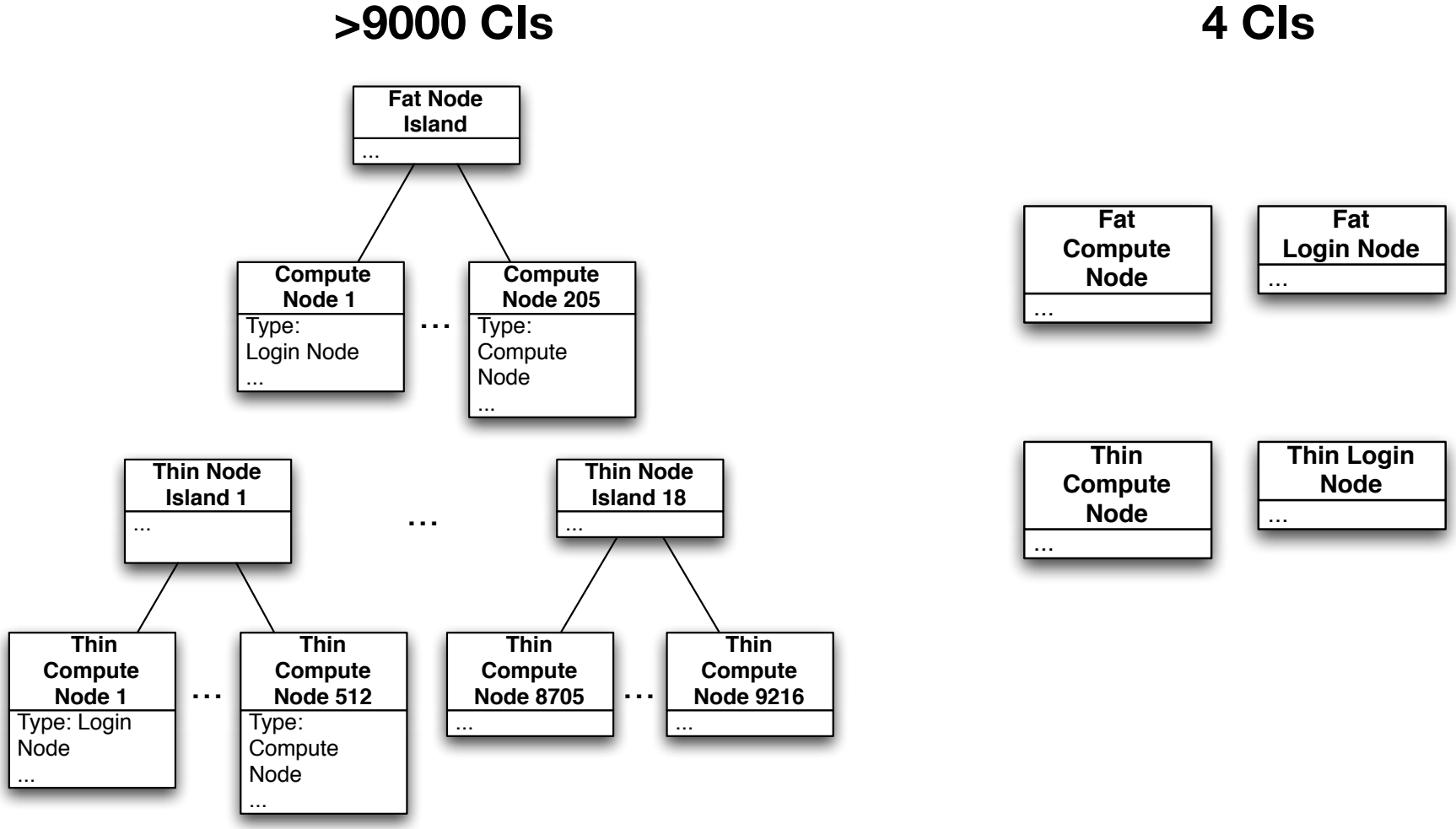
- Idee: Design Patterns verwenden
  - Pragmatischer Mittelweg zwischen „das Rad für jede CMDB neu erfinden“ und „CMDB-Standard-Design“
  - Keine vorgegebenes Modell, sondern Good Practice Lösungsschablonen als Hilfestellung für eigenes Design
- Existierende, beispielhafte Patterns
  - *Collective CI*
  - *Rich CI Relation*
  - *Multi-Value Attributes*
  - Zusammenfassung vieler Diskussionen und Lessons-Learned über die letzten 3 Jahre
  - Letzte Woche auf IEEE/IFIP BDIM-Workshop vorgestellt



MVC Entwurfsmuster für SW-Design

Wie kommt der SuperMUC in die CMDB?

Zwei Arten die SuperMUC-Nodes zu erfassen

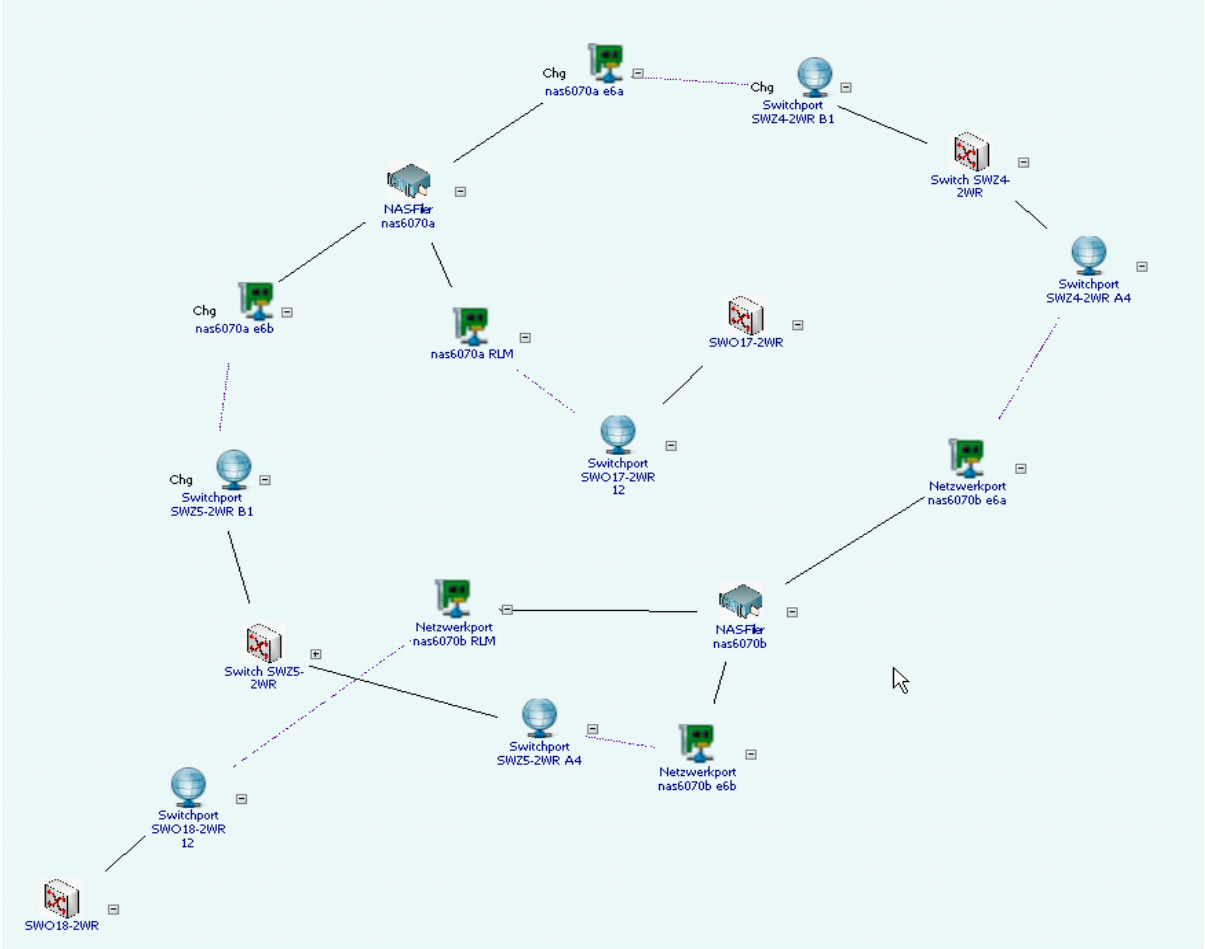


Pattern *Collective CI*

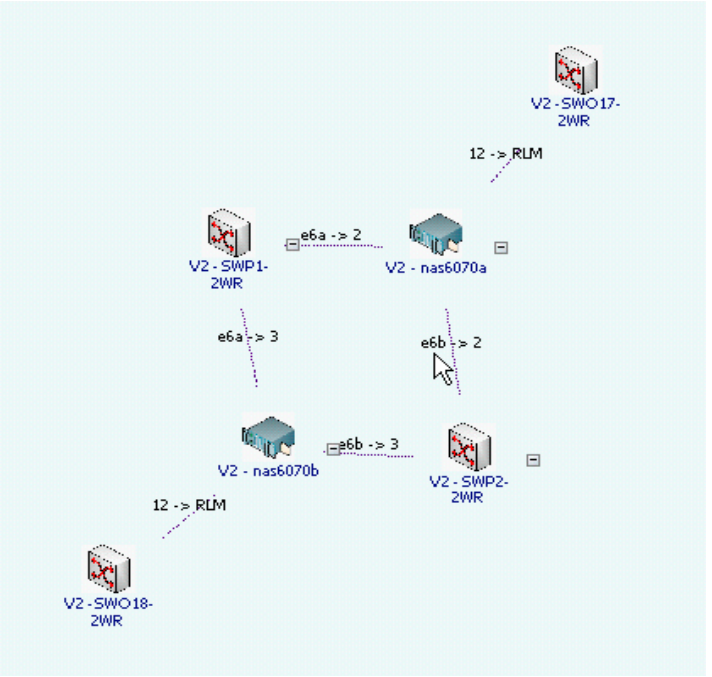
- *Beschreibung*
  Collective CI – Ein einzelnes CI als Platzhalter für viele Komponenten
- *Voraussetzung*
  Eine Gruppe von Komponenten bleibt immer in der (im Scope des Change- bzw. Configuration-Managements) gleichen Konfiguration; immanent durch die Art der Komponente oder durch entsprechende Policy.
- *Vorteil*
  Signifikante Reduzierung der CI-Anzahl CIs, erleichterte Pflege
- *Nachteil*
  Individuelle Relationen und Attribute der zusammen gefassten Komponenten bleiben bleiben in der CMDB undokumentiert

Netzverbindungen

Model 1



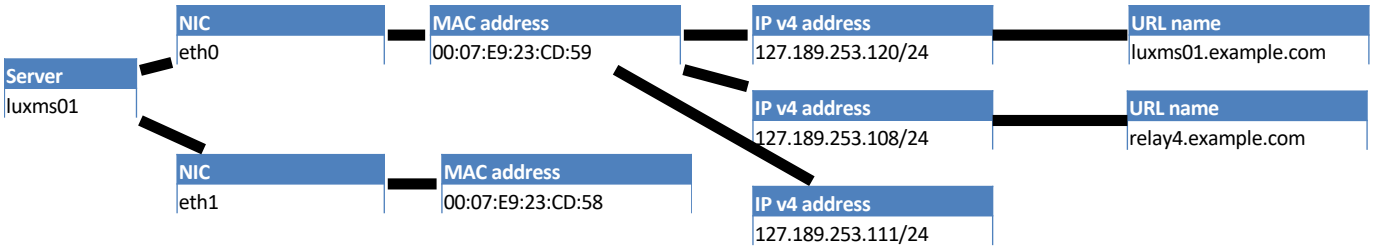
Model 2



Pattern *Rich CI relations*

- *Beschreibung*
  Rich CI relations – Erweiterung von Relationen um editierbare Attribute
- *Voraussetzung*
  Definition von Relationen entsprechend erweiterbar
  (in der Praxis oft durch einfache Erweiterung SQL-Tabellen möglich)
- *Vorteil*
  Signifikante Reduzierung der CI-Anzahl; wichtige Relationen sind leichter erfassbar
- *Nachteil*
  Änderung des Meta-Datenmodells notwendig; etwas aufwändigere Anpassung erforderlich (Entwicklerstudio statt Parameterverwaltung)

Bsp: Komplexe Netzkonfigurationen eines Servers



Ansatz 1:

„CIM-Style“

+ Methodisch sauber
 - sehr viele CIs und Relationen
 - viele Abfragen werden sehr komplex
 - Konsistenz schwer sicher zu stellen

Ansatz 2:

Nahe am typischen out-of-the-box Datenmodell

+ Einfache Abfragen bleiben einfach
- Assoziationen gehen verloren

Server	
Sys-Name	luxms01
MAC	00:07:E9:23:CD:59
MAC	00:07:E9:23:CD:58
IPv4	127.189.253.120/24
IPv4	127.189.253.108/24
IPv4	127.189.253.111/24
IPv4	127.189.253.102/24
IPv4	127.189.253.100/24
IPv4	11.77.6.1/24
IPv4	11.77.6.51/24
URL name	luxms01.example.com
URL name	relay2.example.com
URL name	relay4.example.com
URL name	relay6.example.com
URL name	relay2.mail.example.com

Komplexe Netzkonfigurationen – 3. Ansatz

Ansatz 3:

CI-Attribute von einem „Multi-value“- / Record-Typ

+ Anzahl der CIs und Relationen bleibt relativ gering
+ Zusammenhänge bleiben erhalten
+/- Abfragen werden ein wenig komplexer

Server			
luxms01			
eth0	00:07:E9:23:CD:59	127.189.253.120/24	luxms01.example.com
eth0	00:07:E9:23:CD:59	127.189.253.108/24	relay4.example.com
eth0	00:07:E9:23:CD:59	127.189.253.111/24	-
eth0	00:07:E9:23:CD:59	127.189.253.102/24	relay2.example.com
eth0	00:07:E9:23:CD:59	127.189.253.100/24	relay6.example.com
eth1	00:07:E9:23:CD:58	11.77.6.1/24	-
eth1	00:07:E9:23:CD:58	11.77.6.51/24	relay2.mail.example.com

Pattern *Multi-value Attributes*

- *Beschreibung*
  Multi-value attributes – Record-Typ für CI-Attribute
- *Voraussetzung*
  (Leichter, wenn DBMS oder ITSM-Tool prinzipiell Record-Typen unterstützt, z.B. Unterstützung komma-separierter Listen in VARCHAR.)
- *Vorteil*

  Modellierung komplexer Netz- und Massenspeicher-Konfigurationen mit Erhaltung der Zusammenhänge zwischen dem Unterkomponenten. Z.B.
  - DNS-Name/IP-Adresse/MAC-Adresse/NIC
  - Massenspeicher-Typ/Device-Name/Größe/Mountpoint
- *Mögliche Nachteile*
  I.d.R. tiefere Anpassung an Tool erforderlich (Entwicklerstudio statt Parameterverwaltung).
  Ressourcen (z.B. IP-Adressen, NICs) nicht mehr als eigenes CI nachverfolgbar

Zusammenfassung

- Grundidee der CMDB Patterns:
  Good Practice für das Design von CMDBs
- Die drei vorgestellten Patterns
  - geben bewährten Lösungsmustern einen Namen;
  - dokumentieren Vor- und Nachteile der Anwendung;
  - sind allgemein gehaltene Informationsmodell-Muster;
  - zielen vor allem auf eine Reduktion der Komplexität bei annähernder Erhaltung des Nutzwerts.
- Andere Typen von CMDB Patterns wären möglich, z.B.
  - für die Modellierung konkreter Infrastrukturkomponenten;
  - für die Festlegung des CMDB-Scope;
  - für die Architektur der föderierten CMDB.

Wie geht es weiter?

- Idealvorstellung:
  Diskussion und Ergänzung der Patterns durch eine Community von Configuration-Management-Praktikern und Wissenschaftlern
- Eingang einer späteren Version in einen FitSM-Guide