**Konzept AIR RFC 9022 –**

**Unterstützung aller CI Typen in AIR**

**Content**

1 Einleitung 5

1.1 Neue CI-Typen 5

1.2 Anpassungsbedarf 5

1.3 Spezifikation der Datenfelder 6

2 Anforderungen 6

2.1 Erweiterung der internen Applikationsarchitektur 7

2.2 Anpassungen für komplexe AIR Teilbereiche 7

2.2.1 Advanced Search – Datenbasis 8

2.2.2 Advanced Search – Suchmasken 9

2.2.3 Wizard 10

2.2.4 Relationships 11

2.3 CI Specifics – Hauptbereich der Anpassung 11

2.3.1 IT Items 12

2.3.2 Location Items 12

2.3.3 Organizational Items 13

2.3.4 Einbau der neuen CI-Typ Masken 13

2.3.5 Stammdatenvorhaltung für neue CI-Typ Masken 14

2.3.6 Zusammenfassung der getroffenen Entscheidungen und Roadmap 15

2.4 Erweiterung der Middle-Tier Schicht 15

2.5 Anpassungen weniger komplexer AIR Bereiche 16

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Change | Author | Date |
| 1.0 | Unterstützung aller CI Typen in AIR | S. Pepping (ERCVA) | 2013/01/17 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**List of Figures**

[Figure 1: Overview of the Graphical User Interface 5](#_Toc346177680)

# Einleitung

Ziel dieses Konzeptdokuments ist die Definition aller fachlichen und technischen Anforderungen, für die Einführung aller CI-Typen in AIR zusätzlich zu den bereits verfügbaren "Application", "CommonService", "Middleware" und "Application Platform" der Transbasetabelle "Anwendung" im Rahmen von Master-RFC 9022. In jedem Unterkapitel von Hauptkapitel zwei werden zentrale AIR Bereiche behandelt, die für die erfolgreiche Umsetzung von Master RFC 9022 maßgeblich sind. Es muss für jedes dieser Unterkapitel verbindlich festgelegt werden, welche der möglichen technischen Strategien verfolgt werden soll, um die fachlichen Anforderungen zu erfüllen.

Der Ablauf innerhalb jedes Unterkapitels von Hauptkapitel zwei, an dessen Ende eine Entscheidung gefällt werden muss, ist einheitlich gestaltet. Es beginnt mit einer Darstellung der generellen Thema- und Problemstellung. Um dieses Problem, bzw. die Themenanforderung zu erfüllen, werden Alternativen, bzw. konkurrierende Strategien aufgezeigt. Es wird auch erklärt warum die Themenfrage oder dessen Problem erfolgskritisch ist und welche technischen Aufwände, Vor- und Nachteile hinter jeder Lösungsstrategie stecken.

Dieser Ablauf bildet die Grundlage für die verbindliche Entscheidung.

## Neue CI-Typen

Laut SISECGUI sind die neuen CI-Typen folgende:

* IT Items
  + Hardware System
  + Transient System Platform
  + Net Security Zone
* Location Items
  + Position
  + Room
  + Building Area
  + Building
  + Terrain
  + Site
  + Pathway
* Organizational Items
  + Function

## Anpassungsbedarf

Es gibt durch die neuen CI-Typen Anpassungsbedarf an bereits bestehenden AIR Bereichen. Die folgende Liste definiert für jeden Bereich den Anpassungsbedarf. Bezugspunkt für einen Bereich ist das Navigationsmenu der AIR Anwendung.

|  |  |
| --- | --- |
| **MyPlace** | **Anpassungsbedarf** |
| I’m Owner  I’m Delegate | Ja – Kapitel 2.5  Ja – Kapitel 2.5 |
| **Search** |  |
| Advanced Search  Org Unit Search | Ja – Kapitel 2.2.1, 2.2.2  Ja – Kapitel 2.5 |
| **New & Delete** |  |
| New by Wizard  New by Copy  Delete | Ja – Kapitel 2.2.3  Ja – Kapitel 2.5  Ja – Kapitel 2.5 |
| **View & Edit** |  |
| Details | Ja – Kapitel 2.5 |
| Specifics | Ja – Kapitel 2.3 |
| Contacts | Nein |
| Agreements | Nein |
| Protection | Ja – Kapitel 2.5 |
| Compliance | Ja – Kapitel 2.5 |
| License & Costs | Ja – Kapitel 2.5 |
| Relationships | Ja – Kapitel 2.2.4 |
| Documentation | Ja – Kapitel 2.5 |
| History | Nein – Kapitel 2.5 |

## Spezifikation der Datenfelder

Die Verwaltung der AIR Attribute der CI-Typen der Transbase-Tabelle "Anwendung" werden in der Datei:

[**http://sp-coll-bbs.bayer-ag.com/sites/000127/ConfigMgmt/Forum/non-public/AIRattributeRequirement.xlsx**](http://sp-coll-bbs.bayer-ag.com/sites/000127/ConfigMgmt/Forum/non-public/AIRattributeRequirement.xlsx)

Die Spezifikation der Transbase Attribute der CI-Typen erfolgt in BYTerminal:

[**http://sp-coll-bbs.bayer-ag.com/sites/000127/ConfigMgmt/Forum/BYTerminal(P).xls**](http://sp-coll-bbs.bayer-ag.com/sites/000127/ConfigMgmt/Forum/BYTerminal(P).xls)

Zwecks Einheitlichkeit ist mittelfristig eine Konsolidierung dieser Dokumente vorzunehmen, d.h. die AIRAttributeRequirements.xlsx muss in der BYTerminal(P).xls aufgehen.

# Anforderungen

Es gibt zwei Strategien die Anforderungsblöcke der Unterkapitel umzusetzen. Die erste ist, Kapitel 2.1 bis 2.4 für sich isoliert als Ganzes nacheinander durchzuführen, d.h. jeweils für alle CI-Typen werden alle komplexen AIR Teilbereiche (Kapitel 2.2) erweitert. Anschließend kann der eigentliche Hauptbereich, die CI „Specifics“ für alle CI-Typen (Kapitel 2.3) entwickelt werden.

Stehen alle AIR Oberflächen Bereiche für alle anderen CI-Typen bereit, können die Web Service und Hibernate Komponenten (Kapitel 2.4) erweitert und neu erstellt werden, um die richtigen Transbase Tabellen für die CRUD Operationen anzusteuern.

Für jedes der Unterkapitel ist ein Sub-RFC zu schreiben. [Strategie S1]

Die zweite, alternative Vorgehensweise ist die Umsetzung der Kapitel 2.2 bis 2.4 und deren Sub-RFCs für alle jeden CI Typen einzeln. Für jeden CI-Typ der in AIR neu eingeführt wird werden die Sub-RFCs der Kapitel 2.2 bis 2.4 durchgeführt. Kapitel 2.1 ist prinzipiell nur einmal für alle CI-Typen umzusetzen.

Mit der zweiten Strategie können schneller erste Ergebnisse CI-Typ für CI-Typ sichtbar gemacht und genutzt werden.

Für jeden neuen CI-Typ ist ein Sub-RFC für die Kapitel 2.2 bis 2.4 zu erstellen. [Strategie S2]

**VERBINDLICHE** Entscheidung für: Strategie S1[ ] Strategie S2[ ]

**Entschieden durch Gerling, Klaus (IMGLG) am TT.MM.JJJJ**

## Erweiterung der internen Applikationsarchitektur

Die Erweiterung der internen Applikationsarchitektur- und Logik ist von Grund auf zu monolitisch auf CI-Typen der Tabelle Anwendung ausgelegt gewesen, bevor der Autor die Hauptentwicklung übernommen hat. Um das Einlesen und Updaten der Daten der neuen CI Typen zu gewährleisten, muss dieser Mangel beseitigt werden.

Bisher wurde beim Laden und Speichern der CI-Typen „Anwendung“, „Anwendungsplattform“, „Middleware“ und „Common Service“ nur auf die Transbase-Tabelle „Anwendung“ zugegriffen. Mit der Einführung der neuen CI-Typen ist die Zugriffstabelle für das Laden und Speichern abhängig vom CI-Typ unterschiedlich. Ein zusätzlicher Parameter „tableId“ muss berücksichtigt werden, um auf die richtige Transbase-Tabelle zuzugreifen.

Folgende AIR Komponenten sind in die Verwaltung des „tableId“ Parameters involviert:

* AirBootstrap
* AirApplicationManager
* CiResultGrid
  + MyPlaceTabView
  + CiResultView
  + CiCopyFromView
* CiEditView
* CiSpecificsView

## Anpassungen für komplexe AIR Teilbereiche

Die in diesem Kapitel genannten AIR Bereiche haben hinsichtlich der Unterstützung der neuen CI-Typen eine höhere Komplexität als die anderen Bereiche, die in Kapitel 1.2 in der linken Tabellenspalte aufgeführt sind. Sie werden daher speziell in diesem Kapitel behandelt.

### Advanced Search – Datenbasis

Aus der Sicht des Autors ist die **erste Frage**, die beantwortet werden muss die, gegen welche Tabelle(n) bei der Advanced Search gesucht werden soll. Eine präzise technische Begründung warum, ist ebenfalls verpflichtend. Entscheidend ist diese Frage deshalb, weil sie der Startpunkt für alle weiteren Tätigkeiten bei der CI Bearbeitung ist.

Auch was die Durchführung der weiteren Entwicklungstätigkeiten im Rahmen des Master-RFCs 9022 angeht, ist die Implementierung der neuen CI-Typen bei der Advanced Search die Voraussetzung um mit Kapitel 2.3 fortfahren zu können. Grund hierfür ist, dass erst einmal ein CI eines neuen CI-Typs auswählbar, bzw. erreichbar sein muss, um dessen Datenfelder laden und bearbeiten zu können.

Die evtl. spätere Entscheidung, in einem neuen RFC die C-Eye Suchtabelle als ExtJS Komponente zu extrahieren und in AIR zu verwenden, sollte auch mit eine Rolle bei der Entscheidung spielen, gegen welche Tabelle(n) bei der Advanced Search gesucht werden soll.

Ein weiteres Entscheidungskriterium sind die CI-Typ spezifischen Felder, nach denen je nach CI-Typ gesucht werden können soll. Soll es möglich sein, nach einzelnen CI-Typ spezifischen Feldern zu suchen oder genügt eine Sammlung definierter Standardfelder, die für alle neuen CI-Typen ausreichen müssen?

Die beiden Strategien zwischen denen entschieden werden muss, sind:

[Strategie SAdvS1]: Suche gegen die Tabelle „DWH\_ENTITY“

[Strategie SAdvS2]: Suche gegen die CI-Typ spezifischen Tabellen

[Strategie SAdvS3]: Suche gegen die CI-Typ spezifischen Tabellen und DWH\_ENTITY je nach

zu definierendem Anwendungsfall

Die Suche gemäß SAdvS1 gegen die Transbase-Tabelle „DWH\_ENTITY“ hat den Vorteil, dass unabhängig vom ausgewählten CI-Typ nur eine Tabelle verwendet werden muss, was die Implementierung einfacher macht. Es müssen weder Hibernate Klassen für jeden CI-Typ erstellt werden noch, als Alternative CI-Typ spezifische SQL Statements im Java Code geschrieben werden. Wenn die „DWH\_ENTITY“ Tabelle alle Attribute enthält, mit denen nach allen CI-Typen gesucht werden können soll und dies auch performant, für den Benutzer schnell genug passiert, kann diese Strategie alle Anforderungen abdecken. Der Nachteil ist, dass die bestehende Datenbankfunktionalität der Suche nach CI-Typen der Transbase-Tabelle „ANWENDUNG“ für „DWH\_ENTITY“ umgebaut, bzw. neu implementiert werden muss.

Dem entgegen steht SAdvS2, die in den CI-Typ spezifischen Tabellen sucht. So ist es in SISec realisiert. So wurde auch in AIR bisher verfahren. Die Suche nach Transbase-Tabelle „ANWENDUNG“ CI-Typen geht nicht auf „DWH\_ENTITY“. Diese Strategie sieht vor, weiterhin nicht „DWH\_ENTITY“ zu verwenden, sondern die spezifische Transbase-Tabelle für jeden CI-Typ. Die Implementierung dieser Strategie ist aufwendiger. Sie könnte aber schneller laufen, da in den spezifischen Tabellen natürlich weniger Datensätze vorhanden sind, die die Oracle Datenbank berücksichtigen muss. Ob dem so ist, müsste durch Such- und Performance Tests ermittelt werden.

Strategie SAdvS3 ist ein flexibler Kompromiss, der vorsieht alle neuen CI-Typen über „DWH\_ ENTITY“ gehen zu lassen und die bestehende Funktionalität der Suche nach Transbase-Tabelle „ANWENDUNG“ CI-Typen zu belassen. Ob später die bisherigen CI-Typen auf „DWH\_ENTITY“ umgestellt werden, kann später, unabhängig von diesem RFC oder Konzept, entschieden werden. Es ist sogar möglich, für bestimmte der neuen CI-Typen dessen spezifische Transbase-Tabelle zu verwenden. Diese Flexibilität hat jedoch den Nachteil, dass der Code komplexer wird als bei der ersten Alternative.

**VERBINDLICHE** Entscheidung für:

Strategie SAdvS1[ ] Strategie SAdvS2[ ] Strategie SAdvS3[ ]

**Entschieden durch Gerling, Klaus (IMGLG) am TT.MM.JJJJ**

**Begründung:**

…

Unabhängig von dieser Frage ist die Ermittlung der verfügbaren CI-Typen aus der Transbase. Welche Tabelle, View oder Funktion soll verwendet werden mit evtl. welchen technischen SQL Abfrage WHERE Einschränkungen? Die CI-Typ Ergebnisliste kann mit ExtJS Bordmitteln für die Advanced Search und für die CI Relationships Maske (Kapitel siehe 1.2 und 2.2.3) genutzt werden.

**SQL Abfrage der CI-Typ Liste:**

…

### Advanced Search – Suchmasken

Es gibt für die „Anwendung“ CI-Typen eine Liste von Feldern, die für alle bereits bestehenden CI-Typen relevant sind. Einziger Sonderfall ist der CI-Typ „Application“. Bei diesem kommen zusätzliche Suchfelder hinzu, die wieder verschwinden, sobald ein anderer CI-Typ zur Suche ausgewählt wird. Bei der Einführung der neuen CI-Typen ist die Art und Weise der Unterbringung der neuen Suchfelder in der Advanced Search eine weitere Entscheidung die getroffen werden muss.

Es gibt drei Wege, die neuen Suchfelder der neuen CI-Typen für den Anwender erreichbar zu machen

[Strategie SAdvSMask1]: eine neue Maske pro neuem CI-Typ

* Vorteile
  + Klare Code-Trennung zwischen den CI-Typen und deren Suchfeldern
* Nachteile
  + Mehr Javascript Code mit redundanten Suchfeldern je CI-Typ Suchmaske
  + Längere Ladezeiten beim Anwendungsstart
  + Zusätzlich Schaltlogik zwischen CI-Typ Suchmasken und bei der Suchparameterspeicherung

[Strategie SAdvSMask2]: eine neue Maske für alle neuen CI-Typen

* Vorteile
  + am wenigsten neuer Javascript Code
  + Keine Suchfeldredundanz
  + Minimal wenig verlängerte Ladezeiten beim Anwendungsstart
* Nachteile
  + Hohe Komplexität der einen Suchmaske
  + Komplexität steigt steil an bei mehr inhaltlich nicht zusammenhängende Suchfelder Kategorien. Z. B. wenn Suchattribute bei IT Items und Location Items auf der Maske behandelt werden müssen

[Strategie SAdvSMask3]: eine neue Maske pro CI-Typ Kategorie

* Vorteile
  + Ausgewogenes Verhältnis zwischen neue Codemenge und moderat verlängerter Ladezeiten beim Anwendungsstart
  + Keine Suchfeldredundanz
  + Komplexitätsbegrenzung durch Trennung der CI-Typ Suchfeldkategorisierung
* Nachteile
  + Mehr Schaltlogik bei abhängigen Suchfeldern als bei SAdvSMask1

**VERBINDLICHE** Entscheidung für:

Strategie SAdvSMask1 [ ] Strategie SAdvSMask2 [ ] Strategie SAdvSMask3 [ ]

**Entschieden durch Gerling, Klaus (IMGLG) am TT.MM.JJJJ**

### Wizard

Auch bei der Anlage neuer CIs gibt es zwei Strategien, dem AIR Benutzer zu ermöglichen die Mandatory- und Required- Felder zu bearbeiten. Bisher können ausschließlich nur „Application“ Typ CIs für die Tabelle Anwendung neu angelegt werden. Daher war bisher die Auswahlbox für den CI-Typ des Wizard deaktiviert.

Diese CI-Typ Combobox muss nun aktiviert werden und mit allen CI-Typen ausgestattet werden, wie sie in Kapitel 1.1 definiert sind. Von ihr aus erfolgt die Steuerung, welche CI-Typ spezifische Felder für die Neuanlage gefüllt werden müssen und können („Mandatory“ und „Required“).

[Strategie SW1] sieht vor, für jeden CI-Typ und dessen spezifische „Mandatory“ und „Required“ Felder neue Komponenten/Masken zu entwickeln. Je nach Auswahl des CI-Typs in der dafür vorgesehenen Combobox, wird die entsprechende Wizard CI-Typ Maske eingeblendet.

[Strategie SW2] bedeutet das Verhalten von SISec an dieser Stelle zu übernehmen. Wird ein non Transbase-Tabelle „Anwendung“ CI Typ ausgewählt, würde AIR in die CI Bearbeitungsmasken springen, wie sie in Kapitel 1.2 unter „View & Edit“ aufgeführt sind.

Von dort aus müssen und können die „Mandatory“ und „Required“ Felder bearbeitet und gesichert werden.

Die folgenden Auflistungen zeigen die Vor- und Nachteile der beiden Strategien.

[Strategie SW1]: für jeden CI-Typ eine neue Maske

* Vorteile
  + Führung des Users durch den Wizard
  + Beibehaltung des Wizard Konzeptes im Vergleich zu Strategie SW2
* Nachteile
  + höherer, zusätzlicher Entwicklungsaufwand
  + längere Ladezeiten beim Applikationsstart

[Strategie SW2]: Wechsel, bzw. Umleitung auf die CI-Typ spezifischen „View & Edit“ Masken

* Vorteile
  + keine längere Ladezeiten beim Applikationsstart
  + weniger, zusätzlicher Entwicklungsaufwand
* Nachteile
  + Aufgabe des schon immer vorherrschenden Wizard Prinzips
  + keine Anleitung des Benutzers

**VERBINDLICHE** Entscheidung für: Strategie SW1[ ] Strategie SW2[ ]

**Entschieden durch Gerling, Klaus (IMGLG) am TT.MM.JJJJ**

### Relationships

Die Regeln, welche CI-Typen mit welchen verknüpft werden dürfen, ist in der Relationship-Matrix definiert. Sie enthält bereits alle CI-Typen. Zu finden ist sie unter

[**http://sp-coll-bbs.bayer-ag.com/sites/000127/ConfigMgmt/Forum/non-public/Transbase Relationship-Matrix.docx**](http://sp-coll-bbs.bayer-ag.com/sites/000127/ConfigMgmt/Forum/non-public/Transbase%20Relationship-Matrix.docx)

Die Auswahl der generell verknüpfbaren CI-Typen ist noch beschränkt auf die CI-Typen der Transbase-Tabellen „Anwendung“ und „IT\_System“ mit den Typen „Hardware System“ und „Transient System Platform“.

Die Combobox, die diese CI-Typen enthält, ist auf dieselbe Weise anzupassen wie die des Wizard aus Kapitel 2.2.2. Es muss dieselbe SQL Abfrage verwendet werden. Die PL/SQL Funktion „FT\_RelatedCIs“ liefert laut EVBSI auch schon verknüpfbare CIs zu allen CI-Typen zu einer gegebenen CI-Id. Sie erwartet die Parameter „tableId“, „ciId“, einen Parameter mit den Werten „Upstream“ oder „Downstream“ sowie einen CI Name Parameter um das Suchergebnis einzuschränken.

Weitere Anpassungen sind bis dato nicht notwendig. Die AIR „Upstream“ und „Downstream“ Relationship GUI Tabellen der Seite “Relationships” werten die Relationship-Matrix aus, die bereits die neuen CI-Typen berücksichtigt, sodass an dieser Stelle nichts weiter getan werden muss.

## CI Specifics – Hauptbereich der Anpassung

Die Erweiterung der CI-Typ Maske "Specifics" setzt die technische Umsetzung von Kapitel 2.1 voraus. Die Erweiterung umfasst CI-Typ spezifische Felder wie sie im folgenden Dokument per Screenshot aus SISec erkennbar sind:

[**http://sp-coll-bbs.bayer-ag.com/sites/000127/ConfigMgmt/Forum/non-public/SISec CI-Typ Specifics Masken.docx**](http://sp-coll-bbs.bayer-ag.com/sites/000127/ConfigMgmt/Forum/non-public/SISec%20CI-Typ%20Specifics%20Masken.docx)

### IT Items

|  |  |  |
| --- | --- | --- |
| **CI-Typ :: Transbase-Tabelle** | **AIR-UI CI-Feld** | **Transbase-Feld** |
| System Plattform :: IT\_SYSTEM |  |  |
|  | HW Relation |  |
|  | Group |  |
|  | Type |  |
|  | Name |  |
|  | Cluster Code |  |
|  | Cluster Type |  |
|  | Virtual Hardware Client |  |
|  | Virtual Hardware Host |  |
|  | Virtual Software |  |
|  | Lifecycle |  |
|  | Operational Status |  |
|  | Primary Function |  |
|  | License Scanning |  |
|  |  |  |
| Net Security Zone :: NET\_SECURITY\_ZONE |  |  |
|  | Lifecycle |  |
|  | Operational Status |  |

### Location Items

|  |  |  |
| --- | --- | --- |
| **CI-Typ :: Transbase-Tabelle** | **AIR-UI CI-Feld** | **Transbase-Feld** |
| Position :: SCHRANK |  |  |
|  | Room | RAUM\_ID |
|  | Building Area |  |
|  | Building |  |
|  | Terrain |  |
|  | Site |  |
|  |  |  |
| Raum :: RAUM |  |  |
|  | Floor | ETAGE |
|  | Building Area | AREA\_ID |
|  | Building |  |
|  | Terrain |  |
|  | Site |  |
|  | Country |  |
|  | Street / Number |  |
|  | Postal Code / Location |  |
|  |  |  |
| Building Area :: BUILDING\_AREA |  |  |
|  | Building | GEBAEUDE\_ID |
|  | Terrain |  |
|  | Site |  |
|  |  |  |
| Building :: GEBAEUDE |  |  |
|  | Terrain | TERRAIN\_ID |
|  | Site |  |
|  | Country |  |
|  | Street / Number | STRASSE / NR |
|  | Postal Code / Location | PLZ / ORT |
|  |  |  |
| Terrain :: TERRAIN |  |  |
|  | Site | STANDORT\_ID |
|  | Country |  |
|  |  |  |
| Site :: STANDORT |  |  |
|  | Site | STANDORT\_CODE |
|  | Category | SITE\_CATEGORY |
|  | Plant (SAP) | PLANT\_SAP |
|  | Country | LAND\_ID |
|  |  |  |
| Pathway :: WAYS |  |  |
|  | Connection | CONNECTION |
|  | Endpoint | CI\_A\_TABLE\_ID / CI\_A\_ID |
|  | Endpoint | CI\_B\_TABLE\_ID / CI\_B\_ID |
|  | Vendor | VENDOR\_ID |
|  | Curcuit Name | VENDOR\_CIRCUIT\_NAME |

### Organizational Items

|  |  |  |
| --- | --- | --- |
| **CI-Typ :: Transbase-Tabelle** | **AIR-UI CI-Feld** | **Transbase-Feld** |
| Function :: FUNKTION |  |  |

**… FEHLT NOCH: den Lokationsketten Hintergrund für die Felder erläutern, bei denen kein Datenbankfeld in der rechten Tabellenspalte steht …**

### Einbau der neuen CI-Typ Masken

Es sind zwei Strategien wählbar, wie die Lese- und Bearbeitungsmasken der neuen CI-Typen in das CiEditView Modul integriert werden können:

[Strategie SCIEDIT1]: eine Maske für jeden einzelnen CI-Typ

[Strategie SCIEDIT2]: eine Maske nur für jede CI-Typ Kategorie: IT Items, Location Items und

Organizational Items

SCIEDIT1 bedeutet, dass jedes Feld eines CI-Typs wie es in den Kapiteln 2.3.1 bis 2.3.3 für jeden CI-Typ aufgeführt ist, genauso auf einer CI-Typ Maske platziert wird. Hierbei kommt es zu einem mehrfachen Auftreten gleicher Felder auf den jeweiligen CI-Typ Masken. Z.B. wäre das Feld „Building“ als ID-Name Wertpaar-Combobox auf den Masken der CI-Typen „Position“, „Raum“ und „Building Area“ vorhanden. Für den Browser und den Anwender bedeutet das eine verlängerte AIR GUI Aufbau- und Wartezeit im Vergleich zu SCIEDIT2.

SCIEDIT2 vermeidet das Mehrfachaufkommen von Feldern auf den CI-Typ Masken von SCIEDIT1. Auch die Anzahl neuer Masken ist mit drei deutlich geringer. Jedoch ist die Implementierungskomplexität höher, gerade bei den CI-Typen der Location Items. Dies umfasst die Ein- und Ausblendung von CI-Typ spezifischen Feldern, sowie das Laden und Speichern der dort hinterlegten Daten. Z.B. müssen beim Laden eines „Raum“ CIs mehrere Felder eingeblendet werden, die zuvor ausgeblendet waren, als beispielsweise ein „Building Area“ CI geladen war. Entsprechend muss dies beim Laden und Speichern der Daten von der und auf die Transbase-Datenbank berücksichtigt werden, sodass etwa verhindert wird, dass „Raum“ CI Daten mitgesendet werden, wenn ein „Building Area“ CI gespeichert wird.

**VERBINDLICHE** Entscheidung für: Strategie SCIEDIT1 [ ] Strategie SCIEDIT2 [ ]

**Entschieden durch Gerling, Klaus (IMGLG) am TT.MM.JJJJ**

### Stammdatenvorhaltung für neue CI-Typ Masken

Eine wichtige Frage wenn es um die am benutzerfreundlichste Verteilung von **einmaligen** Datenladezeiten der Stammdaten pro Sitzung geht, die in den Datenfeldern der neuen CI-Typ Seiten benötigt werden, ist der Zeitpunkt, wann im Applikationssitzungsverlauf vom Einloggen bis zum Ausloggen, dieser Ladevorgang stattfindet. Auch die Häufigkeit der Ladevorgänge kann je nach gewähltem Vorgehen variieren.

Auch hier gibt es wieder zwei Wege, die beschritten werden können

[Strategie SLOAD1]: Stammdaten laden: nach dem Login und vor dem UI Aufbau

[Strategie SLOAD2]: Stammdaten laden: On Demand, d.h. beim ersten Aufruf eines non Appli

cation CI-Typ

* Strategie SLOAD1
  + Nachteile
    - längere Ladezeiten beim Start,
    - überflüssiges Laden, wenn nur Application CI-Typen genutzt werden (dürfen, z.B. bei Rolle AIR.Application Layer)
  + Vorteile
    - Alle Daten stehen für alle CI-Typ Masken zu Verfügung und müssen nicht nachgeladen werden
    - Einfachere Implementierung
* Strategie SLOAD2
  + Vorteile
    - keine Verlängerung der Ladezeiten beim Applikationsstart 🡪 Vermeidung der Nachteile der ersten Variante
  + Nachteile
    - Zusätzlich einmalige Ladezeit für Daten von Combobox- und anderen Datenfeldern pro Anwendungsstart bevor CI eines neuen CI-Typs angezeigt werden kann
    - Implementierung aufwendiger

Stammdaten relevante Felder, die mit Comboboxen bedient werden sind

* Location Item CI
  + Raum
  + Building Area
  + Building
  + Terrain
  + Site
* IT Item CI
  + HW Relation
  + Group
  + Type
  + Name
  + Cluster Code
  + Cluster Type
  + Virtual Hardware Client
  + Virtual Hardware Host
  + Virtual Software
  + Primary Function
  + License Scanning

Bei diesen Feldern muss überprüft werden, wie SISec deren Stammdaten ermittelt. Diese Daten können dann entweder direkt in AIR übernommen werden, oder es werden evtl. WHERE Klauseln aus den SISec SQL Statements entfernt, um sie später, in AIR nach Suchkriterien zu filtern. Letzteres hat sich bereits an mehreren Stellen bei der AIR Entwicklung ausgezahlt. Z.B. bei der ITSec Gruppen Auswahl auf der Compliance Seite oder bei der Anwendungskategorie 2 alias „IT Category“ auf der bisherigen Seite „Specifics“ und der „Advanced Search“ Maske. Bei beiden Stammdatenfeldern wurden jeweils alle Datenelemente aus der Transbase Datenbank geladen und innerhalb von AIR Use-Case abhängig gefiltert, um nur die fachlich richtige Teilmenge der Daten anzuzeigen und auszuwählen zu können.

Bei den CI-Typen „Raum“, „Building“ und „Building Area“ soll die Tragweite dieser Enscheidung anhand von Fragen, Antwortmöglichkeiten und deren Konsequenzen weiter erläutert werden.

* Welche UI Feldtypen sollen auf der Location Item CIs Specifics Maske verwendet werden?
* sollen bei CI-Typ "Raum" die Lokationskettenfelder "Building", "Terrain", "Site" als deaktivierte Comboboxen, als Picker Felder, oder als einfache deaktivierte Textfelder gezeigt werden?
  + Comboboxen: Daten werden zwecks schneller Verfügbarkeit vor dem App Start **einmal** pro Sitzung geladen
  + Picker Felder: Daten werden On Demand bei **jeder** Benutzung in einer Sitzung geladen
  + Textfelder: einfache Darstellung der aufgelösten Namen der jeweiligen Location Item CI-Typen. Kein Laden von Stammdaten.

Entscheidend ist diese Frage bei den Feldern "Building" und "Building Area", die sehr viele Daten zu Auswahl haben. Comboboxen stoßen an Grenzen bei großen Datenmengen, sowohl was die Benutzerfreundlichkeit der Auswahl also auch die Ladeperformance betrifft. Bei Pickerfeldern ergeben sich mehr Möglichkeiten durch Filterungs- und Eingrenzungsparametern.

Beim Wizard zum Anlegen von Location Item CI-Typen muss unabhängig von der Entscheidung zwischen Strategie SW1 oder Strategie SW2 aus Kapitel 2.2.3 bedacht werden, dass Textfelder nicht ausreichen für "Building" und "Building Area" CI-Typen. Z.B. wenn für einen "Raum" die "Building Area" oder für eine "Building Area" das "Building" gesetzt werden soll. Hier müssen die entsprechenden Daten für die Auswahl zur Verfügung stehen. Des Weiteren besteht eine Abhängigkeit zwischen den Daten dieser CI-Typen. Wird beispielsweise ein „Building“ ausgewählt, dürfen nur "Building Area" Daten für das gewählte „Building“ zur Verfügung stehen.

Bei der Neuanlage eines "Building" kann das "Terrain" ausgewählt werden. Hier kommt eine Combobox für das "Terrain" eher in Betracht, weil es deutlich weniger "Terrain" Stammdaten gibt, als bei "Building" und "Building Area". Dies hat eine entschärfende Wirkung bei Datenladezeiten und Auswahlschwierigkeit für den Anwender.

**VERBINDLICHE** Entscheidung für: Strategie SLOAD1 [ ] Strategie SLOAD2 [ ]

**Entschieden durch Gerling, Klaus (IMGLG) am TT.MM.JJJJ**

**Anmerkungen:**

…

### Zusammenfassung der getroffenen Entscheidungen und Roadmap

In diesem Kapitel soll von Anforderungs- und Entwicklungsseite gemeinsam der Zusammenhang zwischen den getroffenen Entscheidungen hergestellt werden und der genaue Realisierungsablauf beschreiben werden. Was wird zuerst gemacht, was folgt warum danach, mit dem Ziel eine kleine Roadmap zu erstellen, die auf einen Blick visualisiert wie vorgegangen wird.

Am Start der Road Map steht die Ausdehnung der Advanced Search auf die neuen CI-Typen. Es wurde in Kapitel 2.2.1 Advanced Search die Strategie SAdvS[?] ausgewählt, weil …

…

Die schrittweise Einführung der CI-Typ (Kategorie) spezifischen Masken richtet sich nach fachlicher Dringlichkeit. So werden „Raum“ CI-Typen besonders dringend benötigt, weshalb mit der Entwicklung dieser Location Item CI „Specifics“ Maske begonnen wird.

…

## Erweiterung der Middle-Tier Schicht

Die Anpassungen und Erweiterungen für den backendseitigen Teil des Einlesens und Aktualisierens der Detaildaten der neuen CI-Typen ist Thema dieses Abschnitts. Neben der CI-ID ist für das Ansteuern der richtigen Transbase-Tabellen auch die CI-TableID notwendig. Dies ist in den Sub-RFCs der Kapitel 2.1 und 2.2.1 vorzubereiten.

Welche Felder je CI-Typ aus- und eingelesen werden müssen richtet sich nach den Tabellen von Kapitel 2.3. Da es sich bei diesen Feldern im Wesentlichen um Stammdaten handelt, müssen deren jeweilige Datenbank-IDs und nicht die Werte beim Laden und Speichern verwendet werden.

Die Java Web Service und Hibernate Schicht muss um die neuen CI-Typen erweitert werden. Jeder neue CI-Typ muss mit einer Hibernate Klasse abgebildet werden. Ein Web Service und eine Hibernate Lade und Speicherkomponente entweder für jeden CI-Typ oder jede CI-Typ Kategorie muss auch implementiert werden.

Dies alles sind die Voraussetzungen, um beim Laden und Speichern die Daten aller CI-Typen vom AIR UI aus in die Transbase und zurück zu bringen.

Die genauen technischen Details für die Entwicklung der Hibernate Entityklassen, der Web Services und den Hibernate Lade und Speicherkomponenten werden hier nicht erläutert, da bereits konzeptionell fertig entwickelte Elemente für die CI-Typen der Transbase-Tabelle „Anwendung“ vorliegen. Für die neuen CI-Typen sind diese Elemente als Vorlage zu verwenden.

Zur Datenübertragung zwischen dem Frontend (GUI) und dem WebService werden DTO-Objekte verwendet. Allgemeingültige Attribute, die in jedem CI verwaltet werden, werden in der Klasse BaseDTO veraltet. Die Implementierungen der jeweiligen CI-Typen (Bsp. RoomDTO) erweitern die geerbten Attribute um die - für diesen CI-Typen – spezifischen Attribute.

Im WebService „CiEntityWS“ werden für die CI-Typen jeweils eigene Methoden (hier am Beispiel vom Objekt „Room“) „getRoom“, „saveRoom“, „deleteRoom“ und „createRoom“ erstellt.

Aus dem WebService werden die Methoden aus dem HibernateModul aufgerufen (Beispiel RoomHbn). In diesem Methoden werden dann z.B. noch die Default-Attribute ergänzt oder die erfassten PersonenDaten geprüft und zum Abschluss das Ergebnis, bzw. die erforderlichen Änderungen über das HibernateObjekt „Room“ direkt in die Datenbank übernommen.

### Feldzugriffsberechtigungsermittlung

Analog zum Zugriffsrechteermittlung für CI-Typen der Tabelle „Anwendung“ muss auch für jedes Feld aller CI-Typen …

## Anpassungen weniger komplexer AIR Bereiche

Im letzten Abschnitt dieses Konzepts sollen die Anforderungen der übrig gebliebenen AIR UI Komponenten definiert werden, die nicht in einem eigenen Kapitel behandelt wurden. Diese Komponenten werden aufgelistet und erläutert was in einem Sub-RFC realisiert werden muss.

* I’m Owner und I’m Delegate
  + Es werden CIs aus den Transbase-Tabellen IT\_SYSTEM und ANWENDUNG gefunden. Die PL/SQL Funktion muss erweitert werden um alle CI-Typen, die der Anwender besitzt, zu liefern
* Org Unit Search
  + Setzt Umsetzung von Kapitel 2.2.1 und 2.2.2 voraus. Die verwendete ExtJS Tabellenkomponente mitsamt der darunterliegenden Funktionalität wird auch bei oder OU Search verwendet. Die CI-Typ Combobox muss ebenfalls wie beim Wizard und der Advanced Search um die neuen CI-Typen erweitert werden
* New by Copy
  + Setzt Umsetzung von Kapitel 2.2.1 und 2.2.2 voraus. Die verwendete ExtJS Tabellenkomponente mitsamt der darunterliegenden Funktionalität wird auch beim Kopieren verwendet.
* Delete
  + Änderungen sollten mit einem neuen RFC behandelt werden, nach Master-RFC 9022. Gerade durch die neuen CI-Typen ergeben sich neue/andere Anforderungen, wie Mehrfachlöschungen in Abhängigkeit bestimmter AIR Rollen. Die bestehende Löschfunktion wäre durch den Umfang dieser neuen Anforderungen überfrachtet und würde auch den ursprünglichen Anforderungen von RFC 7935 „Einbau einer Delete-Funktion“ widersprechen. Es werden alle CIs von Transbase-Tabelle IT\_SYSTEM und ANWENDUNG gefunden bei denen der Anwender CI Owner (Delegate) und/oder Application Owner (Delegate) und/oder Application Steward ist
* View & Edit
  + Details
    - andere Felder je nach CI-Typ (Kategorie). Kann noch später definiert werden, wenn die Kapitel 2.3 und 2.4 umgesetzt sind
  + Protection
    - Das Information Class Feld ist nur für Anwendung CI-Typen relevant
  + Compliance
    - Attribute, die es bei Non-Applikationen nicht gibt, müssen wegfallen: dies sind die Checkboxen für die Konzernrichtlinien 1445 und 1775/2059
  + License & Costs
    - für Non- Applikationen CI-Typen nicht relevant. Menupunkt ausblenden und einblenden für CI-Typ Application
  + Documentation
    - für Non - Applikationen CI-Typen nicht relevant. Menupunkt ausblenden und einblenden für CI-Typ Application
  + History
    - kein Anpassungsbedarf für neue CI-Typen, da dies Thema eines anderen RFCs (und Konzepts) wird

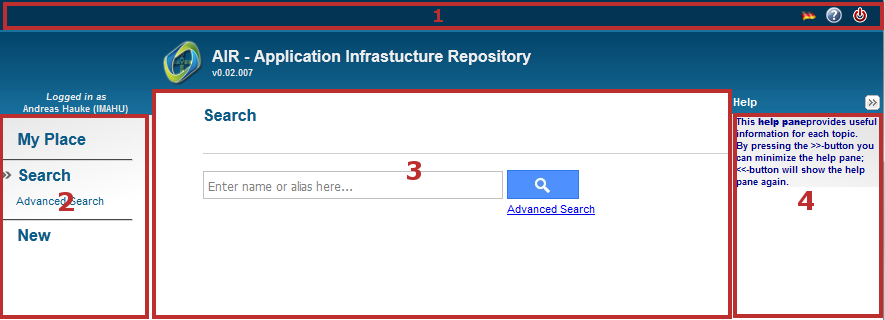


Figure 1: Overview of the Graphical User Interface