# <Beispielsystem>

## *Systementwurf*

Version IsyFact 4.0.x, 14.06.2026

# Inhaltsverzeichnis


Einleitung . . . 3
- Zusammenfassung . . . 4
- Ziel des Dokuments . . . 4
- Leseanleitung . . . 5
- Bezug zum V-Modell® XT . . . 5

Systemüberblick . . . 7
- Systemabgrenzung (Scope) . . . 7
- Einbettung in die Anwendungslandschaft . . . 7
- Randbedingungen und Annahmen . . . 8
- Konformität zur Referenzarchitektur . . . 8

Systemarchitektur des Gesamtsystems . . . 11
- Systemüberblick . . . 11
- Anforderungen . . . 13
- TI-Architektur . . . 14
- Architektur des Teilsystems A . . . 18
- Systemarchitektur des Teilsystems B . . . 31
- Systemarchitektur des Teilsystems C . . . 31
- Querschnittskonzepte . . . 32

Systementwicklung . . . 33

Betrieb . . . 34

Anhang . . . 35
- Anhang A: Optionaler Inhalt . . . 35

Allgemeine Hinweise zur Dokumentvorlage

***Zweck dieses Dokuments***

*Die Dokumentvorlage enthält die Gliederung für einen V-Modell-konformen Gesamtsystementwurf eines Softwaresystems (Software-Entwicklungsprojekt). Es legt fest, welche Inhalte für einen Systementwurf eines IsyFact-Systems festgelegt werden sollen.*

*Der Systementwurf dient dabei folgenden Zwecken:*

- *Grundlage für die Abstimmung mit dem Auftraggeber. Verschiedene Aspekte, wie z. B. die Einbettung in die Gesamtsystemlandschaft, müssen frühzeitig abgestimmt werden.*
- *Anleitung der Entwickler, Vorbereitung auf die Implementierung. Je nach Vorkenntnissen der SW-Entwickler und der Komplexität der umzusetzenden Fachlichkeit müssen diese Vorgaben angemessen detailliert sein.*

*Dieses Dokument dient nicht zur Dokumentation des Systems. Die Systemdokumentation wird separat erstellt, dabei werden Teile dieses Dokuments übernommen.*

***Umfang:***

*In der Regel wird für jedes IT-System (Geschäftsanwendung) jeweils ein eigenes Dokument Systementwurf angelegt. Es gibt jedoch auch IT-Systeme die alleinstehend keine sinnvolle Funktion erbringen können. Dafür kann ein gemeinsamer Systementwurf in einem Dokument erstellt werden.*

*Beispiel:*

*Für ein neues Verfahren sind eine Geschäftsanwendung und ein Service-Gateway zu erstellen. Das Service-Gateway stellt die Services der Geschäftsanwendung bereit. Weiterhin wird ein Native-Client benötigt, welcher ebenfalls die Geschäftsanwendung aufruft. Für diesen Fall werden Geschäftsanwendung, Native-Client und Service-Gateway als Teilsysteme in einem gemeinsamen Systementwurf beschrieben.*

***Nutzung:***

*Sukzessive sollen Ausfüllhinweise und Beispiele in den jeweiligen Kapiteln ergänzt werden. Die Hinweise und Beispiele sind mit einem Rahmen markiert und später zu entfernen.*

*Für die Abstimmungskapitel sind die Inhalte und deren Detailtiefe vorgegeben, für die Anleitung der SW-Entwickler entscheidet der für dieses Dokument verantwortliche Systemarchitekt (Chefdesigner) dabei selber über die Detailtiefe und die konkreten Inhalte.*

***Generelles zur Beschreibung einer Architektur:***

Eine Architektur wird immer Top-down erklärt:

- *Zunächst wird gezeigt, wie sich das neue System in die gesamte technische Anwendungslandschaft einfügt. Kapitel: Systemüberblick/Einbettung in die Anwendungslandschaft.*

- *Danach wird das System detailliert, indem gezeigt wird, wie die Teilsysteme des neuen Systems zusammen arbeiten. Kapitel: Systemarchitektur des Gesamtsystems/Systemüberblick.*
- *Dann wird jedes Teilsystem einzeln beschrieben. Zunächst werden die Komponenten des Systems und ihr Zusammenspiel erklärt, dann wieder die einzelnen Komponenten selbst. Diese schrittweise Vertiefung setzt sich bis zum benötigten Detaillierungsgrad fort.*

*Schnittstellen eines Systems werden immer in der Dokumentation und dem Systementwurf des Implementierers beschrieben. Beim Aufruf einer Schnittstelle eines anderen Systems wird lediglich beschrieben, wie die konkrete Nutzung aussieht, also z. B. wie die internen Daten des Systems auf die Daten der Schnittstelle abgebildet werden.*

*Generell muss in diesem Dokument nichts erklärt und beschrieben werden, was schon in anderen Dokumenten, insbesondere in den Detailkonstruktionen und Vorgaben der Referenzarchitektur enthalten ist.*

***Werkzeugunterstützung bei der Erstellung von UML-Modellen***

*Für die Erstellung der UML- Modelle, die in diesem Dokument als Beispiele aufgeführt sind, wurde das Werkzeug „Enterprise Architect“ der Firma SparxSystems verwendet.*

*Anstelle des Enterprise Architect kann auch jedes andere Modellierungswerkzeug eingesetzt werden, dass den Sprachumfang von UML in hinreichender Breite abdeckt.*

***Systemdokumentation***

*Aus dem Systementwurf entsteht die Systemdokumentation. Die Systemdokumentation erklärt die Zusammenhänge eines Systems und dient als Einstiegspunkt, um den Code zu verstehen und damit Warten und Weiterentwickeln zu können.*

*Generell werden daher zum Zweck der Dokumentation alle Details und Konstruktionshinweise gelöscht, die nur dazu dienten, die Softwareentwickler bei der Programmierung des Systems anzuleiten. In der Systemdokumentation verbleiben alle Teile, die eine Übersicht über die Software beinhalten.*

*In den Kapiteln ist damit folgendes zu modifizieren:*

- ***Kapitel 1**: Anpassen, da sich der Zweck der Systemdokumentation vom dem eines Systementwurfs unterscheidet.*
- ***Kapitel 2**: Belassen.*
- ***Kapitel 3**: Verringern der Tiefe der Beschreibung in den Kapiteln zur Architektur der Teilsysteme und Querschnittskonzepte. Entfernen des Abschnitts zu Anforderungen.*
- ***Kapitel 4**: Die Anpassung dieses Kapitels liegt im Ermessen des Chefdesigners.*
- ***Kapitel 5**: Dieses Kapitel ist in der Regel lediglich eine Referenz. Die Anpassung dieses Kapitels liegt im Ermessen des Chefdesigners.*

# Einleitung

## Vorgehen für die Erstellung des Systementwurfsdokuments

*Das Systementwurfsdokument enthält alle Informationen, welche für die an der Erstellung der Software beteiligten Parteien (Betrieb, Netzwerk-Administration, Auftraggeber, Entwickler, Beschaffung) benötigt werden. Es behandelt eine Vielzahl unterschiedlicher Themenbereiche.*

*Der Detaillierungsgrad, in welchem die Themenbereiche beschrieben werden, ist nicht für alle Systementwürfe gleich: Er hängt von der Art und Komplexität der Problematik und der Erfahrung des Zielpublikums ab. In diesem Dokument werden deshalb beispielsweise keine Vorgaben für den Umfang und die Art der Beschreibung von Anwendungslogik jenseits der Anwendungsfall-Klassen aufgestellt.*

*Auf Grund der obigen Punkte wurde entschieden, das Systementwurfsdokument manuell zu erstellen. Es wurde entschieden, die Teile im Enterprise Architect zu modellieren, bei welchen eine Modellierung im Enterprise Architect möglich und sinnvoll ist. Welche Teile dies sind, wird im vorliegenden Dokument definiert.*

***Vorgaben für die Konstruktion***

*Systementwurf-Dokumente sollen grundsätzlich mit der vorliegenden Vorlage und im Einklang zu den beschriebenen Vorgaben und Ausfüllhinweisen erstellt werden.*

***Modellierung im Enterprise Architect***

*Der Enterprise Architect (oder ein entsprechendes anderes UML-Modellierungswerkzeug) wird für die Modellierung folgender Themenbereiche eingesetzt:*

- *TI-Architektur des Systems*
- *T-Architektur des Systems*
- *Datenmodell*
- *Querschnittskonzepte*
- *Erläuterungen von Lösungsansätzen.*

*Für die obigen Themenbereiche sind jeweils unterschiedliche Diagramme zu erzeugen. Neben den Diagrammen sind vor allem die konkreten Komponenten mit ihren Interfaces, Klassen und Tabellen ein wichtiges Ergebnis der Modellierung im Enterprise Architect. Die Anforderungen an die konkrete Ausgestaltung der Diagramme werden im Rahmen dieser Vorlage in den jeweiligen Kapiteln detailliert.*

*Die Paketstruktur des Enterprise Architect Modells orientiert sich sowohl an der Benamung als auch an der Hierarchie der Verzeichnisstruktur des Systementwurfs. So wird die Einheitlichkeit der EA-Modelle für unterschiedliche Projekte erhöht und das Auffinden benötigter Diagramme erleichtert. Das folgende Listing zeigt beispielhaft ein EA-Modell.*

*Paketstruktur eines Enterprise Architect Modells*

```
📄 Geschäftsanwendung XYZ
  📄 Systementwurf
    📄 Systemarchitektur des Gesamtsystems
      📄 Querschnittkonzepte
      📄 Systemüberblick
      📄 TI-Architektur
    📄 <<IT-System>> Geschäftskomponente XYZ
      📄 Anwendungskern
        📄 <<A-Komponente>> Auskunft
        📄 <<T-Komponente>> Berechtigung
        📄 <<A-Komponente>> Meldung
        📄 <<A-Komponente>> Protokoll
      📄 Batch
      📄 Datenmodell
      📄 Externe Schnittstellen
      📄 Teilsystem-Querschnittskonzepte
      📄 Teilsystem-Überblick
    📄 <<IT-System>> Service-Gateway
    📄 Rückbezug Systemspezifikation
      📄 Anwendungsfälle
      📄 Anwendungskomponenten
      📄 Batches
      📄 Entitäten
      📄 Nachbarsystemschnittstellen
```

# Zusammenfassung

*Kurze Zusammenfassung. Was ist der Hintergrund dieses Systems?*

*Wozu dient dieses System?*

*Gegebenenfalls kann die Zusammenfassung aus der Systemspezifikation übernommen werden.*

# Ziel des Dokuments

Dieses Dokument dient folgenden Zwecken:

- Es ist die Grundlage zur Abstimmung zwischen dem Projektmanagement, der Softwareentwicklung des Auftraggebers/–nehmers und dem Systembetrieb.
- Es dient als Vorgabe und zur Anleitung für die SW-Entwickler.

- Es ist Grundlage für die Erstellung und frühzeitige Abstimmung des IT-Sicherheitskonzeptes (siehe: hier ggf. einen Link setzen).

# Leseanleitung

*An wen richtet sich das Dokument? Z.B. Entwickler, Betrieb, ...*

*Für welche Zielgruppe sind welche Kapitel hauptsächlich interessant?*

# Bezug zum V-Modell® XT

Dieses Dokument ist eine Adaption (Tailoring) der Vorgaben des V-Modells auf die Entwicklung mit der IsyFact. Der Systementwurf kann nach V-Modell® allgemein verschiedene Produkte umfassen. Bei der Nutzung der IsyFact sind einige dieser Produkte bereits vollständig oder in großen Teilen vorgegeben, so dass sie in diesem Dokument nicht mehr beschrieben werden müssen.

Generell sind grundsätzliche Architekturprinzipien und Entwurfsalternativen bereits im Rahmen der Erstellung der IsyFactReferenzarchitektur diskutiert worden und müssen daher in diesem Dokument nicht wiederholt werden. Gleiches gilt für die Absicherung des Designs.

Im Folgenden sind die Produkte eines Systementwurfs gemäß V-Modell® XT aufgeführt und kommentiert.

- **Systemarchitektur:** Die grundsätzliche Systemarchitektur ist durch die Referenzarchitektur vorgegeben. In diesem Dokument wird beschrieben, wie diese Referenzarchitektur instanziiert wird.
  - Die Dekomposition des Systems ist im Kapitel Systemarchitektur des Gesamtsystems beschrieben.
  - Querschnittliche Systemeigenschaften sind bereits im Rahmen der IsyFact-Referenzarchitektur betrachtet worden und werden nicht wiederholt. Sollte eine Verfeinerung oder spezifische Ergänzung der Referenzarchitektur notwendig sein, erfolgt dies im Kapitel Querschnittskonzepte.
  - Die Schnittstellenübersicht des Systems erfolgt der Übersichtlichkeit halber auf zwei Abstraktionsebenen: auf Ebene der Gesamtanwendungslandschaft erfolgt dies im Kapitel Systemüberblick/Einbettung in die Anwendungslandschaft. Die konkreten Schnittstellen des Systems und ggf. von Teilsystemen sind im Kapitel Systemarchitektur des Gesamtsystems/Systemüberblick beschrieben. Die Details zur Implementierung und Dateninhalten der Schnittstellen finden sich in der Konstruktion der jeweiligen Services unter Architektur des Teilsystems A/Service.
  - Die zu spezifizierenden Systemelemente sind der Gegenstand von Kapitel Systemarchitektur des Gesamtsystems, insbesondere in den Kapiteln Architektur des Teilsystems (ff.).
- **Unterstützungs-Systemarchitektur:** Unterstützungssysteme sind in der Regel durch die IsyFact vorgegeben. Sollte es trotzdem nötig sein, ein neues Unterstützungssystem zu konstruieren, erfolgt dies in einem eigenen Dokument nach dieser Vorlage.

- **Mensch-Maschine-Schnittstelle (Styleguide):** Ein Styleguide ist durch die IsyFact nicht vorgegeben. Es wird jedoch vorausgesetzt, dass ein solcher Styleguide für die Organisation, die die IsyFact einsetzt, bereits vorhanden ist und verwendet werden kann.
- **HW-Architektur:** Die HW-Architektur ist als Referenzarchitektur bereits vorgegeben. Die konkrete HW-Architektur für das System befindet sich im Kapitel TI-Architektur.
- **SW-Architektur:** Die SW-Architektur ist in den Grundlagen durch die IsyFact Referenzarchitektur vorgegeben. Die Instanziierung dieser Architektur befindet sich in Abschnitt Architektur des Teilsystems A (ff.).
- **Datenbankentwurf**: Der Datenbankentwurf findet sich unter Datenmodell.
- **Implementierungs-, Integrations- und Prüfkonzepte für System, Unterstützungssystem, HW und SW:** Die Prozesse sind im Rahmen der IsyFact bzw. im Gesamtprojektrahmen vorgegeben und werden hier nicht wiederholt.
- **Migrationskonzept:** Für Migrationsszenarien wird ein eigenes Konzept erstellt. Dieses Dokument macht keine Aussagen zu einer Migration.

# Systemüberblick

## Systemabgrenzung (Scope)

*Was sind die Aufgaben des Systems, was nicht? Nur ein kurzer Abriss.*

*Z.B.: „Verwalten der Daten der Domäne X, keine Bereitstellung der Oberfläche, das erfolgt durch das System Y.“*

## Einbettung in die Anwendungslandschaft

*Wie fügt sich das neue IT-System in die Anwendungslandschaft ein?*

*An dieser Stelle einfügen: Ein Bild der Anwendungslandschaft gemäß Ist- bzw. Referenzarchitektur mit hervorgehobenem neuem System.*

*In jedem Systementwurf soll hier der Bezug zur Referenzarchitektur hergestellt werden. Folgende Inhalte sind aufzunehmen:*

- *Benennung der Teile/Komponenten der Referenzarchitektur, die im Rahmen des Projektes/der Systementwicklung entwickelt werden („Was ist der Beitrag des Projekts zur Referenzarchitektur?“)*
- *Aussagen zur Konformität der Bestandteile / Komponenten des neuen Systems mit der IsyFact-Referenzarchitektur und anderen Vorgaben (z.B. Standard Kataloge für Produkte). Insbesondere: Welche Komponenten sind nicht konform zur Referenzarchitektur und warum nicht?*

*Zur Einordnung in die Systemlandschaft kann eine Abbildung ähnlich der folgenden verwendet werden. Für konkrete Entwürfe muss die Grafik durch eine nicht anonymisierte aktuelle Version der zugehörigen Systemlandschaft ersetzt werden.*

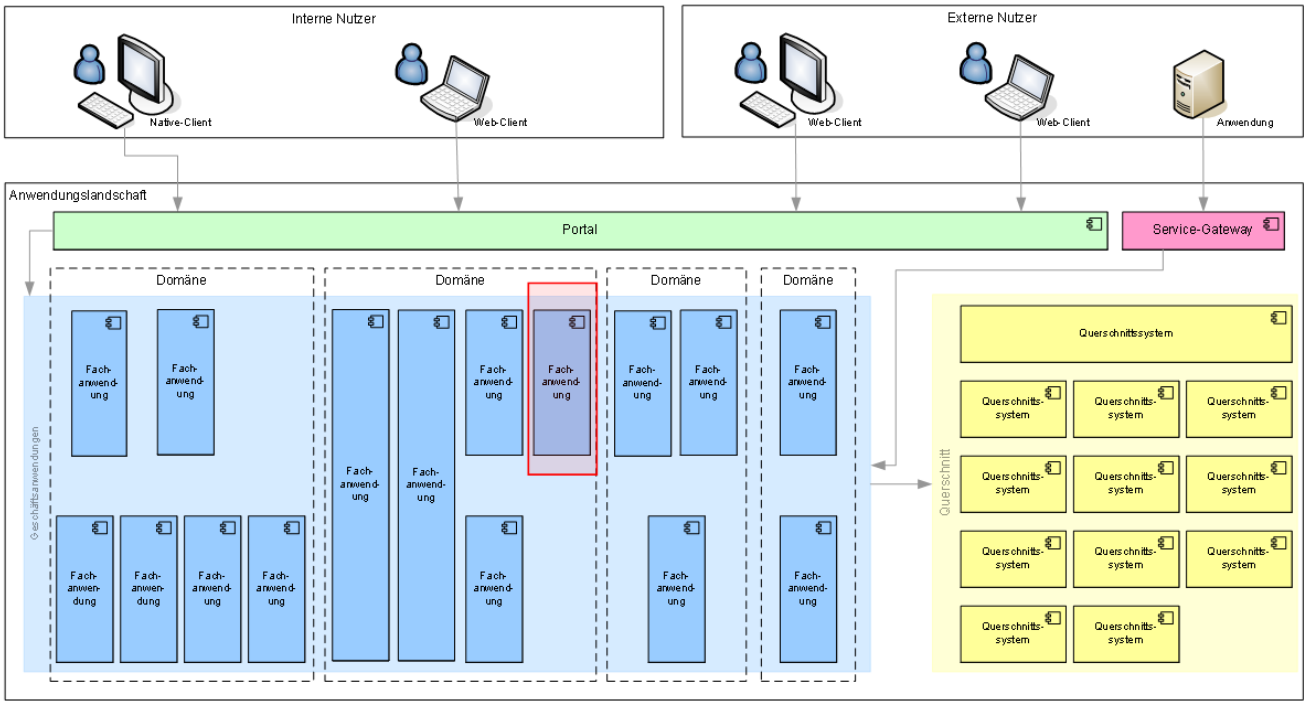


*Abbildung 1. Einbettung von <Anwendung (roten Rahmen platzieren)> in die Systemlandschaft.*

# Randbedingungen und Annahmen

*Verweis auf die Randbedingungen und Annahmen in der Systemspezifikation.*

*Ausführliche Beschreibung von zusätzlichen Randbedingungen und Annahmen, die für den Systementwurf relevant sind.*

# Konformität zur Referenzarchitektur

*In diesem Kapitel wird die Konformität des Systems zur Referenzarchitektur bestätigt und evtl. vorhandene Abweichungen beschrieben und begründet. In dieser Vorlage wird zunächst davon ausgegangen, dass die Standard-Versionen des aktuellen IsyFact Releases verwendet werden. Eine Übersicht über die Software-Artefakte der IsyFact findet sich in Tabelle 1.*

*Für nicht relevante Elemente wird in die Status-Spalte „**nicht benötigt**“ eingetragen.*

*Bei relevanten Elementen, die erweitert werden müssen, um den Anforderungen des Systems gerecht zu werden, wird in die Status-Spalte „**wird erweitert**“ eingetragen. Die durchzuführenden Erweiterungen sind in der darauffolgenden Tabelle zu begründen.*

*Bei relevanten Elementen, die in einer anderen als der in Tabelle 1 angegebenen Version verwendet werden, wird in die Status-Spalte „**abweichende Version**“, in der Version-Spalte die Version des Elements und in der Referenz-Spalte die Referenz auf das Dokument ergänzt. Die Abweichungen sind ebenfalls in der darauffolgenden Tabelle zu begründen.*

Das IT-System <System> wird konform zur Referenzarchitektur der IsyFact umgesetzt. Die folgende

Tabelle führt die vom IT-System verwendeten Elemente der IsyFact auf.

*Tabelle 1. Verwendete Elemente der IsyFact (Standards und Erweiterungen)*

| Thema | Status | Version | Referenz |
|---|---|---|---|
| **Allgemein** | | | |
| IsyFact-Referenzarchitektur | berücksichtigt | 4.0 | referenzarchitektur:index.pdf |
| **Referenzarchitektur** | | | |
| Anwendungskern | berücksichtigt | *Standard* | |
| Batch | berücksichtigt | *Standard* | |
| Persistenz | berücksichtigt | *Standard* | |
| Service | berücksichtigt | *Standard* | |
| Interne Servicekommunikation | berücksichtigt | *Standard* | |
| **Bausteine (Standard)** | | | |
| Fehlerbehandlung | berücksichtigt | *Standard* | |
| Logging | berücksichtigt | *Standard* | isy-logging:konzept/master.pdf, isy-logging:nutzungsvorgaben/master.pdf |
| Sicherheitskomponente | Abweichungen | *Standard* | |
| Überwachung und Konfiguration | berücksichtigt | *Standard* | |
| **Bausteine (Erweiterungen)** | | | |
| Alphanumerisches Suchverfahren | berücksichtigt | *Standard* | |
| Behördenverzeichnis | berücksichtigt | *Standard* | |
| Benutzerverzeichnis | berücksichtigt | *Standard* | |
| Bildbearbeitung | berücksichtigt | *Standard* | |
| Biometrie | berücksichtigt | *Standard* | |
| Mail-Gateway | berücksichtigt | *Standard* | |
| Nummernkreis | berücksichtigt | *Standard* | |
| Output-Management | berücksichtigt | *Standard* | |
| Portal | berücksichtigt | *Standard* | |
| Protokollierung und Protokollrecherche | berücksichtigt | *Standard* | |
| Regelwerk | berücksichtigt | *Standard* | |
| Schlüsselverzeichnis | berücksichtigt | *Standard* | |
| Service-Gateway | berücksichtigt | *Standard* | |

| Thema | Status | Version | Referenz |
|---|---|---|---|
| Spooling | berücksichtigt | *Standard* | |
| Bedienkonzept | berücksichtigt | *Standard* | |
| Umgang mit Sonderzeichen | berücksichtigt | *Standard* | |
| **Plattform** | | | |
| Deployment-Konzept | berücksichtigt | *Standard* | |
| **Methodik** | | | |
| Template Systementwurf | berücksichtigt | *Standard* | |
| Java-Programmierkonventionen | berücksichtigt | *Standard* | |

Abweichungen zur Referenzarchitektur sind in Tabelle 2 dokumentiert und begründet.

*Tabelle 2. Abweichungen zur Referenzarchitektur*

| Thema | Abweichung | Begründung |
|---|---|---|
| Batch | Im Batch XXX (siehe Kapitel XXX) wird… | Der Batch hat extrem große Datenmengen zu bearbeiten. Daher muss… |
| … | | |

# Systemarchitektur des Gesamtsystems

*In diesem Kapitel wird ein Systemüberblick gegeben und die Aspekte des Gesamtsystems beschrieben, die für alle oder mehrere Teilsysteme gelten. Das können Verfahren, Festlegungen oder die Beschreibung gemeinsam genutzter Komponenten sein. In der Regel hat ein System mehrere Teilsysteme, z. B. eine Geschäftsanwendung und ein Service-Gateway.*

*Festlegungen, die bereits in den übergeordneten Dokumenten (z.B. Referenzarchitektur, Sicherheitsrichtlinie, ...) gemacht wurden, sind im gesamten Kapitel nur zu referenzieren, Abweichungen hiervon sind zu begründen.*

## Systemüberblick

***Zweck***

*Dieser Abschnitt beschreibt, aus welchen Teilsystemen das Gesamtsystem besteht und gibt damit einen Top-level-Überblick über das System. Teilsysteme sind z. B.: ein Service-Gateway oder eine Geschäftsanwendung. Dieses Kapitel ist die nächste Konkretisierungsstufe nach der Einordnung in die Anwendungslandschaft aus Kapitel Systemüberblick/Einbettung in die Anwendungslandschaft*

***Inhalte***

*An dieser Stelle befindet sich eine Übersichtsgrafik, die darstellt, aus welchen Teilsystemen das Gesamtsystem besteht, wie diese sich gegenseitig aufrufen und welche Nachbarsysteme und Querschnittsfunktionen aufgerufen werden. Zur besseren Orientierung sind der Farbcode und das Layout der Referenzarchitektur zu beachten. Für die Schnittstellen und Services soll ein geeigneter Abstraktionsgrad gewählt werden. Zweck dieser Übersicht ist es einen prinzipiellen Überblick zu geben und nicht jedes Detail zu beschreiben.*

*In der Beschreibung der Grafik ist Folgendes beschrieben:*

- *Was sind die Aufgaben der einzelnen Teilsysteme, welche Funktionalität bieten sie an?*
- *Wie ist das Zusammenspiel der Teilsysteme um die Gesamtaufgabe des Systems zu erfüllen, wie sieht die Aufgabenverteilung aus?*

*Ein Beispiel für ein solches Diagramm ist:*

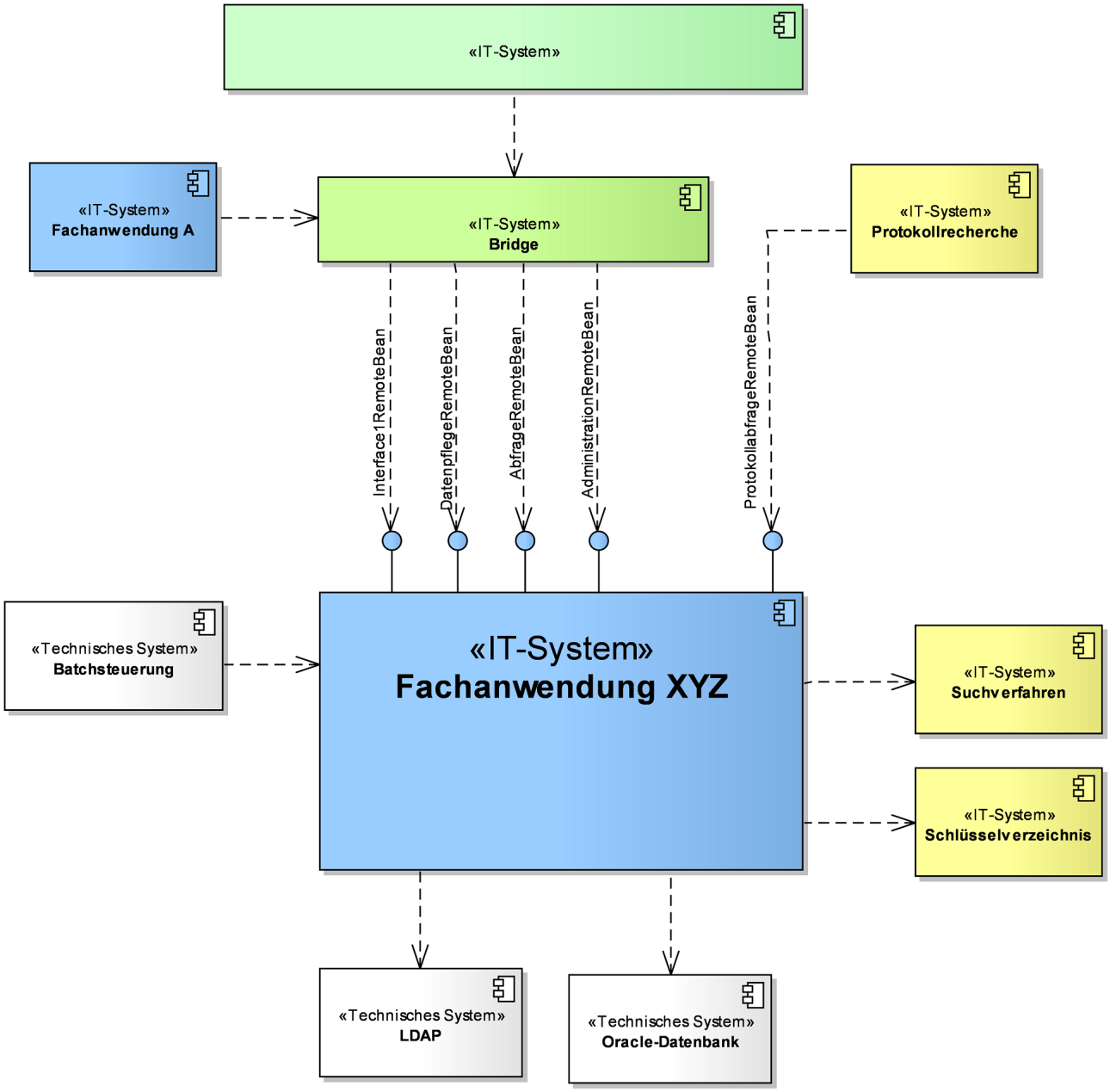


*Abbildung 2. Teilsysteme*

*Die Farbgebung für ein Teilsystem-Element orientiert sich an der fachlichen Referenzarchitektur. Für die verschiedenen Systeme sind folgende Hintergründe zu verwenden:*

- *Blau für Geschäftsanwendungen (R/G/B: 153 / 204 / 255)*
- *Orange für Register (R/G/B: 255 / 204 / 153)*
- *Gelb für Querschnittsanwendungen (R/G/B: 255 / 255 / 153)*
- *Violett für das Service-Gateway (R/G/B: 255 / 153 / 204)*
- *Hellgrün für Portal-Systeme (R/G/B: 204 / 255 / 204)*
- *Weiß für technische Systeme oder Produkte (Datenbank etc.) (R/G/B: 255 / 255 / 255)*

# Anforderungen

*In diesem Kapitel werden die Anforderungen an das System genannt. In der Regel wird auf die Systemspezifikation verwiesen. Abweichungen zur Systemspezifikation sollten aufgeführt werden.*

*Falls Anforderungen seit der Spezifikation hinzugekommen sind oder konkretisiert wurden, wird das hier dokumentiert.*

## Funktionale Anforderungen

*Referenz auf die Systemspezifikation, ein Satz: „Die funktionalen Anforderungen sind in der Systemspezifikation XYZ in Abschnitt n beschrieben…“*

## Nichtfunktionale Anforderungen

*Nichtfunktionale Anforderungen, die schon in der Systemspezifikation beschrieben sind, werden hier nicht wiederholt, sondern nur referenziert und falls erforderlich im Folgenden konkretisiert.*

### Mengengerüst

*Aufrufhäufigkeit von Funktionen, Anzahl von Datensätzen, …*

*- - > Kann tabellarisch dargestellt werden.*

### Verfügbarkeit

*Angabe der Ausfallzeit in Stunden pro Woche und am Stück.*

#### Performance

*Auflistung durchschnittlicher Antwortzeiten und maximaler Laufzeiten für Batches. Bewährt hat sich eine Darstellung mit folgenden Inhalten:*

*Eine Grundlast durch den Ablauf definierter Anwendungsfälle wird auf das System gelegt. Dann wird an einem speziellen Client die Antwortzeiten für einen oder mehrere Anwendungsfälle oder wichtige Funktionen gemessen.*

*Dabei wird die Antwortzeit z. B. so beschrieben: „In 90% der Fälle benötigt die Anfrage weniger als 2s, in 10% der Fälle unter 5s.“ Ggf. kann man noch eine Festlegung aufnehmen, ob einmalige Ausreißer erlaubt sind.*

#### Skalierbarkeit

*In der Regel ein kurzer Satz:*

*Durch die Verwendung der IsyFact-Referenzarchitektur erfüllt die Anwendung XYZ die Anforderungen, die allgemein an Geschäftsanwendungen gestellt werden: Sie läuft auf separaten Hardware-Systemen, die Failover und Lastverteilung ermöglichen. Eine weitere Skalierung ist durch Hinzufügen weiterer Systeme möglich.*

### Sicherheit

*In der Regel in der Spezifikation beschrieben. Verweis auf Sicherheitskonzept*

## TI-Architektur

*Zunächst ein Überblick: Kurze Beschreibung der wichtigsten Eckpunkte der TI-Architektur.*

*TI-Diagramme sind als Deployment Diagramme umzusetzen. Diese enthalten die Elemente:*

- *Component für Software-Deploymenteinheiten*
- *Execution Environment für Application Server, Servlet Container etc.*
- *Nodes für Server, Clients, Router, Firewalls, Loadbalancer, etc.*
- *Boundaries für Netzsegmente*

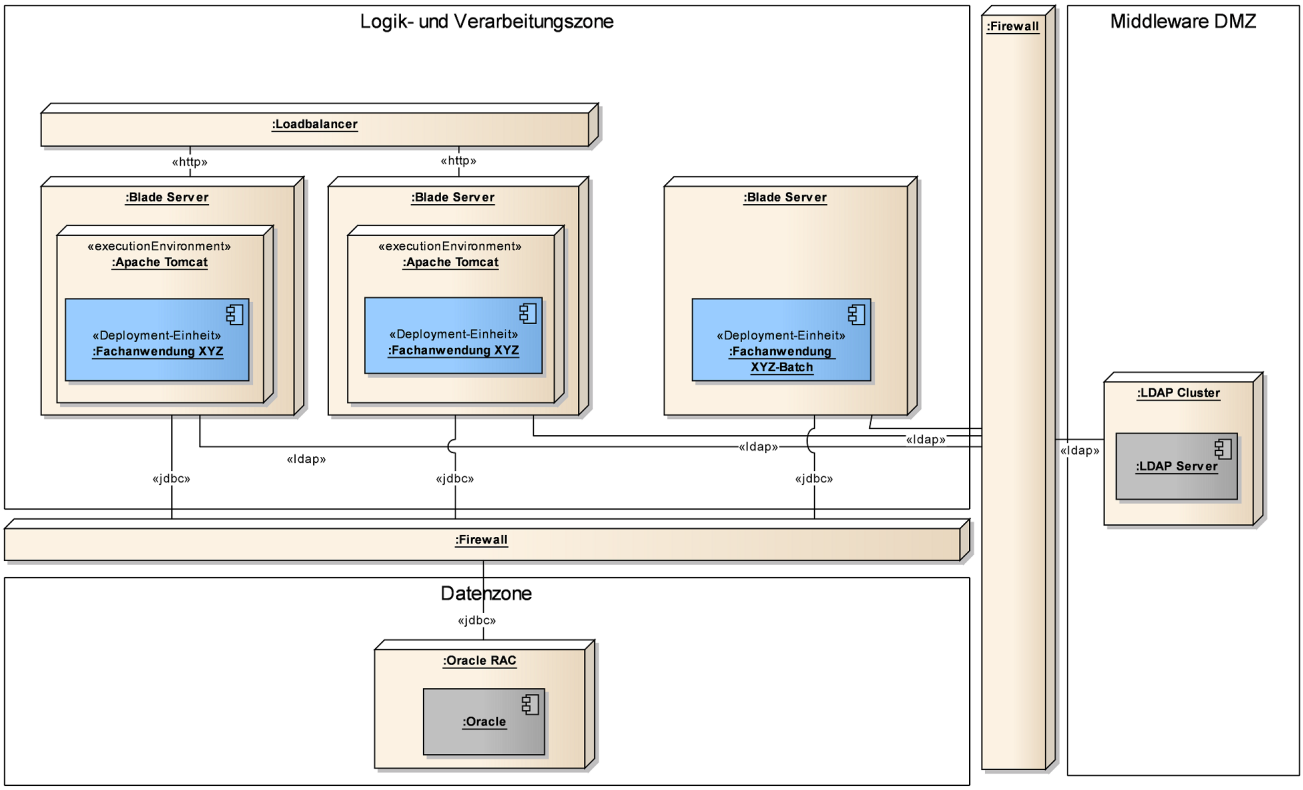


*Abbildung 3. Deployment Diagramm*

*Wichtige Inhalte:*

- *Zuordnung der Server bzw. Laufzeitumgebungen zu den Saga-Zonen.*
- *Zuordnung der Deployment-Einheiten auf die Server/Laufzeitumgebungen.*
- *Kommunikationsbeziehungen werden zu den Loadbalancern bzw. Firewalls dargestellt. Es soll klar werden, über welche Zonengrenzen hinweg mit welchen Protokollen kommuniziert wird.*
- *Existierende Nachbarsysteme werden in der Regel nicht dargestellt. Ausnahmen sind aber möglich, wenn ein besonderer Aspekt dargestellt werden soll. Z.B. ein neuer Native-Client, der an einen bestehenden Service angebunden wird.*
- *Keine konkreten Adressen, Portnummern, Dimensionierungen, sondern eine logische Sicht.*

## Laufzeitumgebung

*Liste der eingesetzten Produkte für die Laufzeitumgebung. Diese Umgebung ist für Anwendungen nach Referenzarchitektur vorgegeben, deshalb kann der Abschnitt unten in der Regel übernommen werden, gegebenenfalls ergänzt um das konkrete Linux-Betriebssystem.*

*Die Versionen sind nach dem Produktkatalog zu überprüfen und mit dem Gesamtprojekt abzustimmen. Für die Laufzeitumgebung wird ein Default-Tomcat bereitgestellt.*

*Die Laufzeitumgebung für das IT-System <SYSTEM> besteht gemäß Produktkatalog der*

*Referenzarchitektur aus den folgenden Produkten:*

*Tabelle 3. Laufzeitumgebung für <SYSTEM>*

| Kategorie | Name | Version | Bemerkung |
|---|---|---|---|
| Betriebssystem | Linux 64 bit | Kernel >= 4.12 | |
| Java-Laufzeitumgebung | OpenJDK (Eclipse Temurin von Adoptium) | 17.x | |
| Servlet-Container | Apache Tomcat | 16.0.18 | |

## Ressourcen

*Abschätzung der Ressourcen in der Produktionsumgebung:*

*- Mengengerüst für die Datenbank*

*- Platzbedarf von (temporären) Dateien*

*Form: Tabellarische Darstellung und erläuternder Text.*

Mengengerüst für die Datenbank

In Tabelle 4 ist eine Abschätzung für die Datenbankgröße enthalten. Hier ist allerdings nur die Größe der Rohdaten angegeben. Für eine Berechnung der Größe der Table-Spaces müssen noch Spezifika der konkret verwendeten Datenbank berücksichtigt werden. Die Berechnung der Größe der Table-Spaces erfolgt durch die Datenbank-Administration auf der Basis der hier genannten Zeilenanzahl je Tabelle.

*Tabelle 4. Mengengerüst für die Datenbank*

| Tabelle/Index | Typ | Anzahl/Zeilen | Größe Rohdaten | | Bemerkung |
|---|---|---|---|---|---|
| | | | Zeile | Tabelle | |
| ... | | | | | |
| ... | | | | | |
| ... | | | | | |

Platzbedarf für Dateien

*Abschätzung der Ressourcen in der Produktionsumgebung:*

*- Mengengerüst für die Dateien auf dem lokalen Laufwerk*

*Form: Tabellarische Darstellung und erläuternder Text.*

In Tabelle 5 ist eine Abschätzung für die auf dem lokalen Laufwerk gespeicherten Dateien enthalten.

*Tabelle 5. Mengengerüst für die Dateien auf dem lokalen Laufwerk*

| Datei | Größe | Bemerkung |
|---|---|---|
| … | | |
| … | | |

## Bibliotheken, Drittsoftware

*Auflistung der benötigten Bibliotheken für die Anwendungsentwicklung und Software auf den Zielsystemen.*

*tabellarische Darstellung*

Das IT-System <SYSTEM> benötigt gemäß Produktkatalog der Referenzarchitektur die folgenden Bibliotheken bzw. Drittsoftware:

*Tabelle 6. Benötigte Bibliotheken und Drittsoftware*

| Kategorie | Name | Version | Bemerkung |
|---|---|---|---|
| Programmiersprache | | | |
| Webservice-Framework | | | |
| Komponenten-Framework | | | |
| Persistenz-Framework | | | |
| Logging-Framework | | | |
| JDBC-Treiber | | | |
| Unit-Testing | | | |
| Scheduler | | | |
| Datenbank | | | |
| … | | | |

# Architektur des Teilsystems A

*Bei der Beschreibung des Gesamtsystems in 3.1 wurden Teilsysteme eingeführt. Diese werden jetzt in eigenen Unterkapiteln der Reihe nach beschrieben. Ggf. kann es sinnvoll sein, ein Kapitel voranzustellen, das Gemeinsamkeiten aller Teilsysteme erklärt.*

*Dieses Kapitel enthält vornehmlich eine Top-down Beschreibung des Teilsystems. Die Beschreibung umfasst die Aufteilung des Systems in Komponenten und deren Bezug zur Systemspezifikation. Wie detailliert die einzelnen Komponenten und deren Teile beschrieben werden, hängt davon ab, wie detailliert die Erläuterungen für die Entwickler sein sollen.*

*Weiterhin werden in diesem Kapitel auch übergeordnete Designprinzipien und Patterns vorgestellt, die bei der Konstruktion der Software eine Rolle gespielt haben.*

*Schließlich finden sich in diesem Kapitel noch Details, auf die der Chefdesigner besonders hinweisen möchte, und die sonst im Code versteckt sind, z. B. die Konfiguration zur Anbindung von Nachbarsystemen.*

***Beschreibung der Teilsysteme***

*Die Teilsysteme haben, wenn es sich um Geschäftsanwendungen handelt, folgende, fest vorgegebene T-Architektur:*

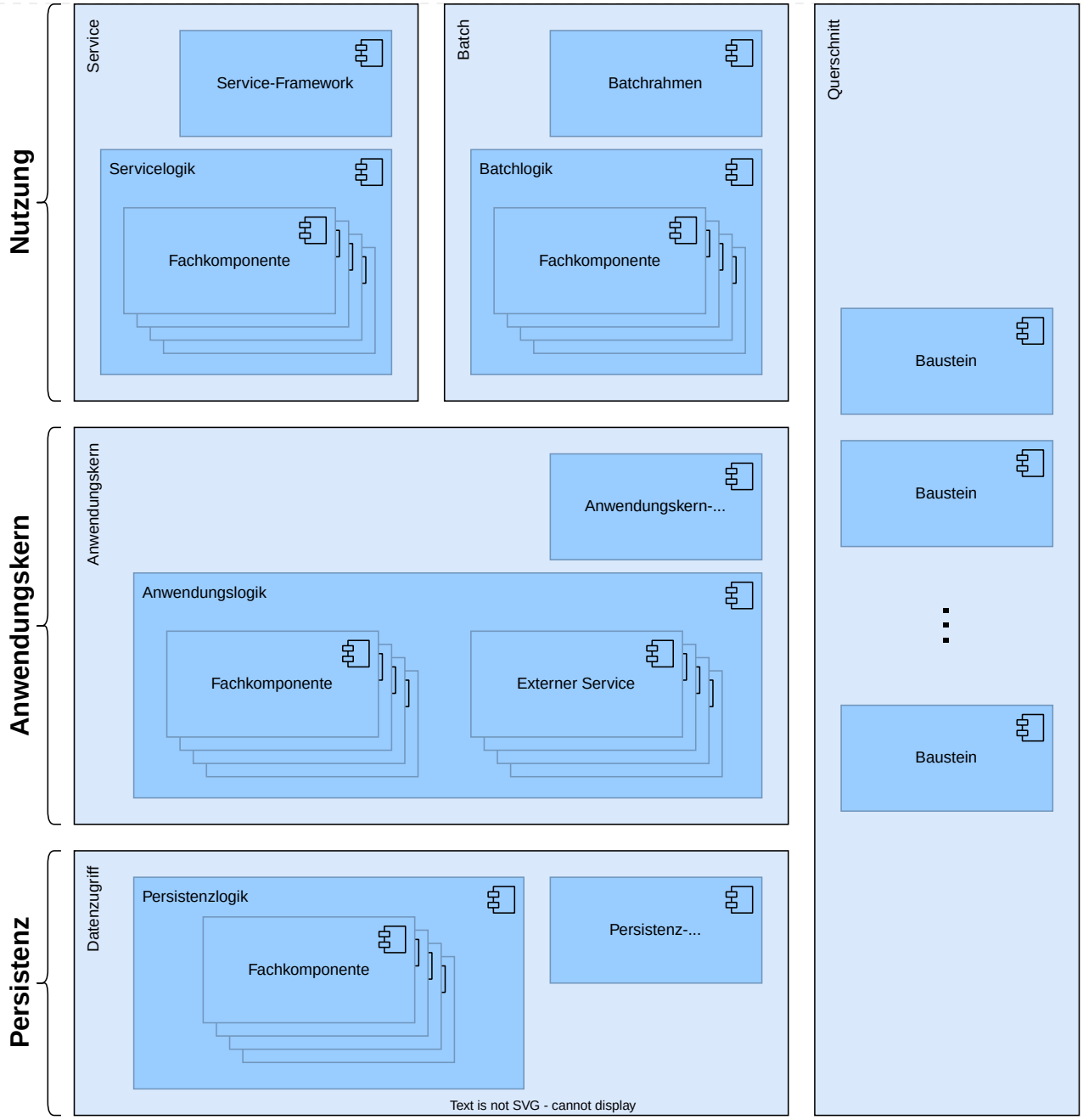


*Abbildung 4. Technisches Architektur eines Teilsystems*

*Die konkrete Ausgestaltung dieser Architektur mit den konkreten Komponenten ist Inhalt dieses Unterkapitels.*

*Dazu wird zunächst eine Übersicht über das Teilsystem gegeben, anschließend werden bei Bedarf die einzelnen Komponenten detailliert.*

*Falls es sich bei dem Teilsystem z. B. um ein Service-Gateway oder einen Querschnittlichen Service handelt, muss die Gliederung im Sinne des oben gesagten. angepasst werden.*

## Überblick

*In diesem Abschnitt findet sich ein Überblick über das Teilsystem. Dies erfolgt mit einem UML-Komponentendiagramm und einem Text der die einzelnen Komponenten und ihr Zusammenspiel erklärt.*

*Das Diagramm soll eine Übersicht über die Schnittstellen, Batches und Komponenten des Teilsystems bieten. Die Zusammenhänge und Beziehungen zwischen den Bestandteilen des Teilsystems sollen dargestellt werden.*

*Das Diagramm soll nicht die Logik der Verarbeitung beschreiben.*

***Dargestellte Elemente***

- *Schnittstellen als Interface-Elemente, Schnittstellen-Klassen als Class-Elemente.*
- *Batches als Class Elemente, Eingabedateien als Artifact-Elemente.*
- *Schichten-Begrenzungen (Batch etc.) als Boundary Elemente*
- *Technische Systeme und Produkte als Component-Elemente*
- *Komponenten-Schnittstellen als Interface-, Komponenten als Component-Elemente.*
- *Für Aufrufbeziehungen ist kein Stereotyp zu verwenden. Keine Zusatzinformationen tragende Stereotypen wie << use >> sind explizit zu vermeiden.*
- *Für Eingabedateien ist das Stereotyp Eingabedatei zu verwenden.*

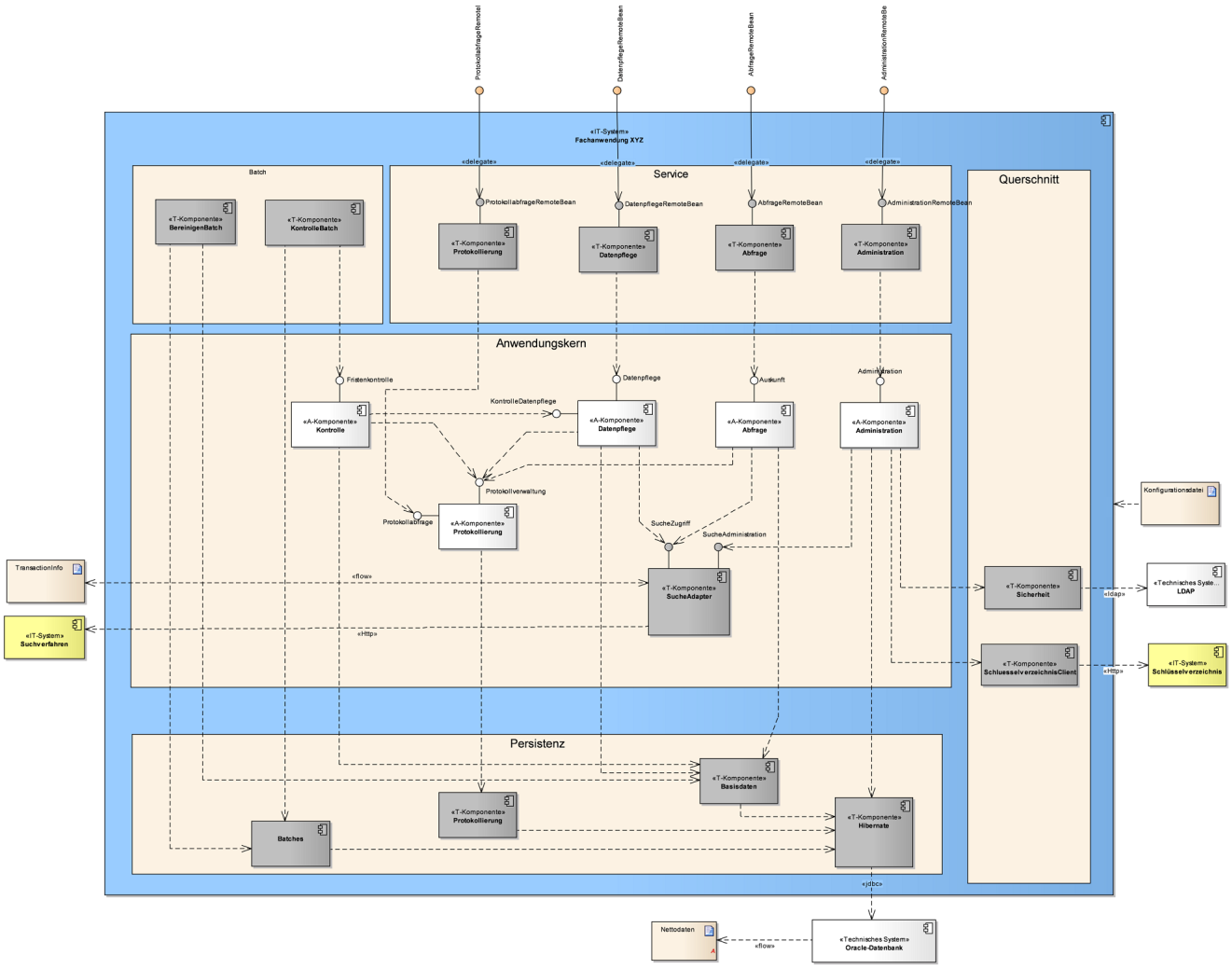


*Abbildung 5. Überblick über die Teilsysteme*

*Die Struktur des Anwendungskerns findet sich in den Komponenten Batch, Service und Persistenz*

*wieder. Der Designer muss abwägen, ob eine differenzierte Darstellung dieser Komponenten hier einen Nutzen bringt.*

## Anwendungskern

*In diesem Abschnitt werden die Komponenten des Anwendungskerns vorgestellt und beschrieben. Die Tiefe der Darstellung richtet sich nach der Komplexität der umzusetzenden Fachlichkeit und den Kenntnissen der Entwickler, die nach diesen Vorgaben die Software programmieren sollen.*

*Jede Komponente wird in einem eigenen Abschnitt vorgestellt.*

**Komponente X**

*In der Regel finden sich hier ein UML-Komponentendiagramm und ein erklärender Text.*

*Dieses Diagramm beschreibt den Aufbau einer Komponente. Es gibt einen Überblick über die wichtigsten Klassen der Komponente.*

***Darstellung***

- *Die Komponente selbst als Component-Element.*
- *Das Interface der Komponente als Interface-Element.*
- *Die Parameter-Klassen und Exceptions des Interfaces als Class- und Interface-Elemente.*
- *Die Klassen der Komponente als Class-Elemente*
- *Nutzungsbeziehungen zwischen den dargestellten Elementen als Dependency-Verbinder.*
- *Implementierungs- und Vererbungsbeziehungen als Realization-Verbinder.*
- *Ein Boundary-Element für die Schicht „Anwendungskern“.*

*Da sich die Komponenten-Innensichten je nach Komponente stark unterscheiden können, werden keine Stereotypen vorgegeben. Der Stereotyp << use >> ist nicht zu verwenden. Die Nutzung folgender Stereotypen für Verbinder wird empfohlen:*

- *Für eine Beziehung, welche über Spring konfiguriert wird: SpringDependency.*
- *Für Datenfluss-Beziehungen (etwa Eingabe-Dateien): flow.*

*Die Komponente wird im Diagramm als Component-Element dargestellt. Die Klassen der Komponente werden innerhalb des Komponenten-Elements dargestellt. Auf jeden Fall dargestellt werden die Fassaden-Klasse sowie die Anwendungsfall-Klassen.*

*Das Komponenten-Interface mit seinen Operationen wird oberhalb des Komponenten-Elements angezeigt. Dargestellt werden außerdem die Parameter-Klassen der Interface-Operationen und die geworfenen Exceptions. Es werden keine Beziehungen zwischen dem Interface und den Parameter-Klassen bzw. Exceptions dargestellt.*

*Spring Beans werden durch den Stereotype „Spring Bean“ gekennzeichnet werden.*

*Alle Elemente werden innerhalb von Package-Elementen dargestellt.*

*Sehr Umfangreiche Datenstrukturen für Ein- und Ausgaben können bei Bedarf in weiteren Diagrammen dargestellt werden.*

*Beispiel für das Diagramm:*

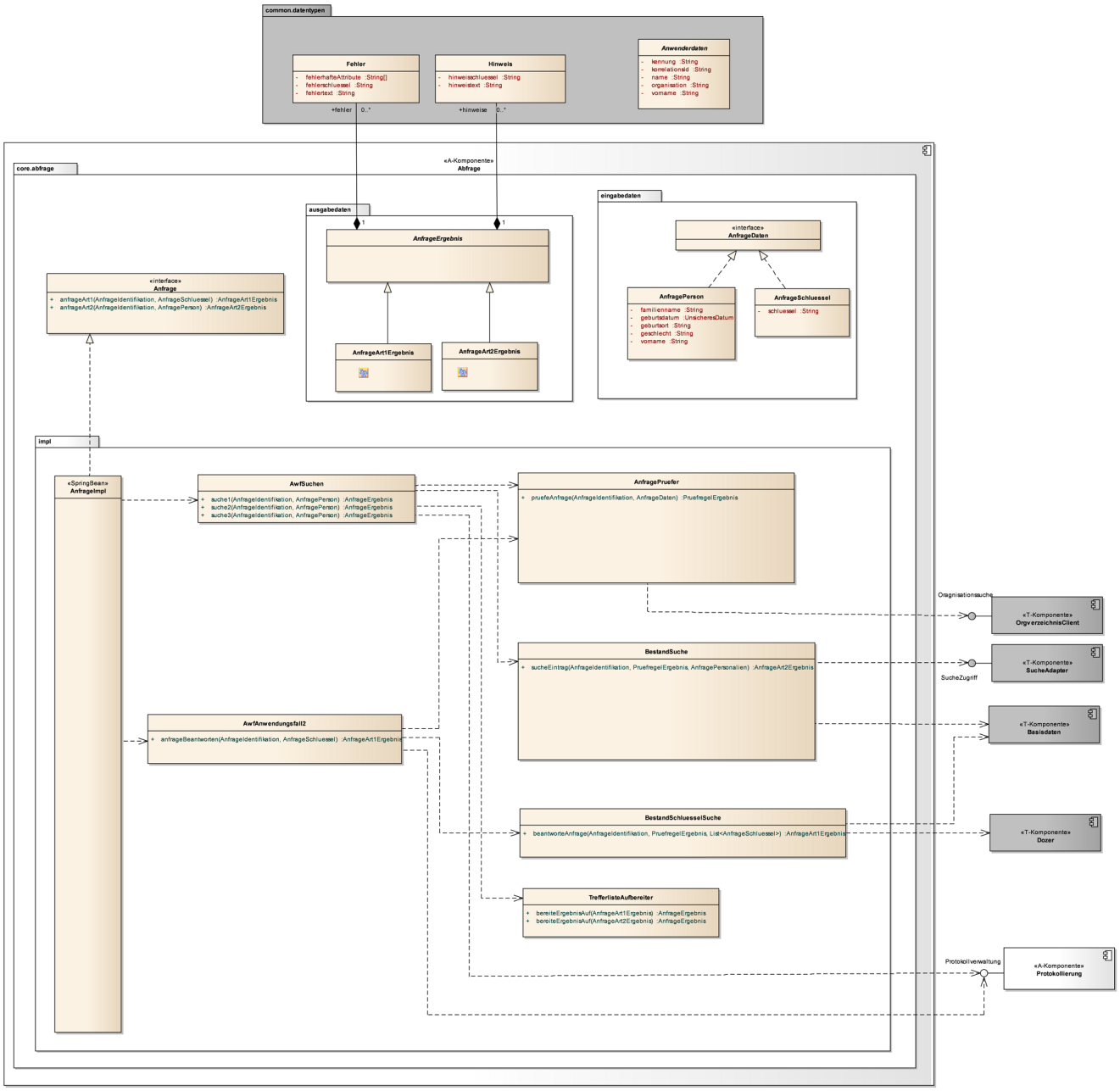


*Abbildung 6. UML-Komponentendiagramm*

*Hier ist sowohl die innere Struktur der Komponente als auch die Beziehung zu aufrufenden/aufgerufenen Komponenten dargestellt.*

***Rückbezug Systemspezifikation***

*Bei der Beschreibung der Komponente muss noch der Bezug zur Spezifikation erklärt werden, d. h. es muss gezeigt werden*

- *wie sich die Anwendungsfälle der Spezifikation auf die Anwendungsfallklassen des Codes abbilden.*
- *wie sich die Anwendungsfunktionen der Spezifikation auf die Anwendungsfunktionsklassen im Code abbilden.*

*Falls die Elemente der Systemspezifikation nicht im Enterprise Architect-Repository vorliegen,*

*werden sie im Diagramm-Paket erstellt. Sie werden nicht dokumentiert und besitzen keine weiteren Eigenschaften. Es sind lediglich Platzhalter.*

*Dargestellt werden:*

- *Die Elemente der Systemspezifikation*
- *Die Elemente der Konstruktion*
- *Die Zusammenhänge als Assoziationen*
- *Eine Boundary, welche die Elemente der Systemspezifikation beinhaltet.*

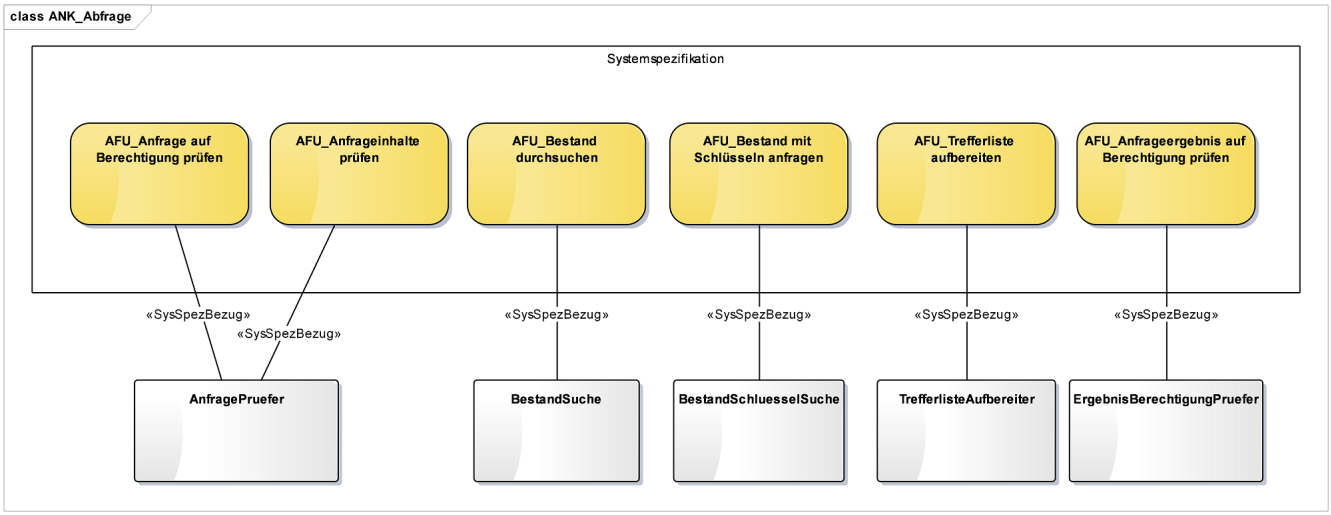


*Abbildung 7. Komponenten Zusammenhänge*

### Komponente Y

### Komponente Z

## Persistenz

*Die Vorgaben zur Persistenz sind in einem eigenen Konzept der IsyFact beschrieben und werden hier nicht wiederholt.*

*In diesem Abschnitt können sich Themen finden wie z. B.*

- *Besonderheiten in der Persistenz, die über die Vorgaben der Referenzarchitektur hinausgehen*
- *Umgang mit Sperren*
- *Ggf. die Konstruktion der persistenten Entitäten und DAOs*

## Batch

*Die allgemeinen Vorgaben zum Batch sind in einem eigenen Konzept der IsyFact beschrieben und werden hier nicht wiederholt. Ggf. werden hier Spezialisierungen und zusätzliche Festlegungen zu den Batches beschrieben.*

*In diesem Abschnitt findet sich die konkrete Konstruktion der Batches, in der Regel mit einem eigenen Abschnitt pro Batch.*

*Weiterhin wird hier der Rückbezug zur Systemspezifikation hergestellt, indem die Batches der Spezifikation den Batches der Software zugeordnet werden. Dies kann textuell oder durch eine UML-Grafik geschehen.*

## Service

*Die allgemeinen Vorgaben zu Services sind in eigenen Konzepten der IsyFact beschrieben und werden hier nicht wiederholt. Ggf. werden hier Spezialisierungen und zusätzliche Festlegungen zu Services beschrieben.*

*In diesem Abschnitt ist die konkrete Konstruktion der einzelnen Services beschrieben, in der Regel mit einem eigenen Abschnitt pro Service, den das System anbietet. Dies sind Services, die per REST bereitgestellt werden. Services, die externen Nutzern zur Verfügung gestellt werden, werden im Rahmen der Dokumentation des Teilsystems Service-Gateway beschrieben. Die eigentliche Servicedokumentation für externe Nutzer ist dabei dann ein eigenes Dokument.*

***Rückbezug Systemspezifikation***

*Bei der Beschreibung der Komponente muss noch der Bezug zur Spezifikation erklärt werden, d. h. es muss gezeigt werden*

- *wie sich die Anwendungsfälle der Spezifikation auf die Anwendungsfallklassen des Codes abbilden.*
- *wie sich die Anwendungsfunktionen der Spezifikation auf die Anwendungsfunktionsklassen im Code abbilden.*

*Falls die Elemente der Systemspezifikation nicht im Enterprise Architect-Repository vorliegen, werden sie im Diagramm-Paket erstellt. Sie werden nicht dokumentiert und besitzen keine weiteren Eigenschaften: Es sind lediglich Platzhalter.*

*Dargestellt werden:*

- *Die Elemente der Systemspezifikation*
- *Die Elemente der Konstruktion*
- *Die Zusammenhänge als Assoziationen*
- *Eine Boundary, welche die Elemente der Systemspezifikation beinhaltet.*

## Auswertungen

*In diesem Abschnitt werden die einzelnen Auswertungen beschrieben und deren Umsetzung. Für Reports, die über ein DWH-Werkzeug bereitgestellt werden, wird es in einer der nachfolgenden Versionen der IsyFact allgemeine Vorgaben geben. Bis dahin sind die spezifischen Vorgaben für das vorliegende System in diesem Abschnitt zu beschreiben.*

## Druck

*In diesem Abschnitt wird die Erstellung der einzelnen Druckstücke beschrieben. Da die Druckstücke durch das Oputput Management erstellt werden, reduziert sich dies auf:*

- *Die Vorlagen*
- *Die Aufbereitung der Daten für die einzelnen Druckstücke*

## Querschnittskonzepte des Teilsystems

*In diesem Abschnitt werden Querschnittskonzepte beschrieben, welche nur für das vorliegende Teilsystem gelten. Falls es keine gibt, entfällt dieser Abschnitt.*

## Datenmodell

*Beschreibung des logischen und physischen Datenmodells des Teilsystems. Die Datenmodelle werden als UML-Klassendiagramme erstellt.*

**Vorgehen zur Konstruktion**

*Die Entitäten des Datenmodells sollen den Komponenten des Anwendungskerns zugeordnet werden. Dazu werden in die Diagramme Boundaries eingezeichnet, die die Entitäten einer Komponente einrahmen.*

*Das Vorgehen zur Konstruktion der Entitäten ist dabei in der Regel das folgende:*

1. *Aus dem fachlichen Modell (UML-Modell im Modellierungswerkzeug) werden die Entitäten in das technische Modell übernommen. Dieses ist ebenfalls ein UML-Modell. In der Regel entsprechen die Entitäten des technischen Modells 1:1 den Entitäten des fachlichen Modells, daher ist diese Übernahme ein guter Ausgangspunkt.*
2. *Das technische Modell wird angepasst, indem z. B. Denormalisierungen vorgenommen werden oder Entitäten mit 1:1-Relationen in einer Entität zusammengefasst werden.*
3. *Aus dem technischen Modell im Modellierungswerkzeug werden dann die entsprechenden Java-Klassen für die Entitäten generiert.*
4. *Die generierten Java-Klassen werden mit Annotationen für das DB-Mapping ergänzt.*
5. *Aus diesen annotierten Klassen wird mit Hilfe der entsprechenden Hibernate-Werkzeuge das passende DB-Schema generiert.*
6. *Dieses Schema wird als physisches Datenbankschema in das Modellierungswerkzeug importiert. Damit erhält man eine Darstellung der Entitäten, die die Grundlage für die Grafiken zum Datenmodell im Systementwurf sind.*
7. *In der Regel ist das so entstandene Datenmodell zu groß und unübersichtlich, um in einer einzigen Grafik dargestellt zu werden. Wenn möglich wird das Diagramm daher in Teil-Diagramme zerlegt, die in einer logischen Reihenfolge aufeinander aufbauen. Grundsätzlich sollen sich diese Teildiagramme am Komponentenschnitt orientieren, sie können aber auch noch feiner zerlegt werden. Zusätzlich zu den einzelnen Teil-Diagrammen sollte auch noch das Gesamt-Datenmodell in Form einer Übersicht (z. B. ohne Attribute) dargestellt werden, um dem Leser eine bessere Orientierung zu geben.*

**Logisches Datenmodell**

*Das logische Datenmodell entsteht aus dem fachlichen Datenmodell der Systemspezifikation. Es ist weitgehend identisch, passt jedoch z. B. die Namensgebung der Entitäten den Coding-Richtlinien an und legt fest, in welcher Richtung Relationen navigierbar sind.*

*Das Diagramm ist eine Übersicht über das logische Datenmodell eines Teilsystems. Es enthält alle persistenten Entitäten des Systems.*

***Darstellung***

- *Beziehungen zwischen Entitäten werden über Assoziationen dargestellt.*
- *Dargestellt werden Entitäten über Class-Elemente mit Stereotyp „Entität“.*
- *Es werden alle Attribute dargestellt. Besondere technische Attribute für primär Schlüssel und optimistisches Locking werden über die Stereotypes „Id“ und „Version“ gekennzeichnet.*
- *Die Komponenten, zu denen eine Gruppe von Tabellen gehört, werden über Boundary-Elemente dargestellt.*
- *Die Beziehungen zwischen Tabellen werden über Assoziation-Verbinder dargestellt.*
- *Die Tabellen sollen nach Komponenten gruppiert werden. Jede Komponente soll als Boundary um die Tabellen der Komponente dargestellt werden.*

*Innerhalb einer Komponente sollen die folgenden Regeln für Tabellen eingehalten werden:*

- *Voneinander erbende Tabellen sollen untereinander (von oben nach unten) dargestellt werden.*
- *Dekomposition von Tabellen soll von links nach rechts dargestellt werden: Bestandteil-Tabellen stehen rechts von ihren Haupt-Tabellen.*

*Die Wichtigkeit von Entitäten soll über ihre Position und Größe dargestellt werden. Das Ziel ist ein intuitiv möglichst gut verständliches Diagramm zu erstellen, in dem z.B. die Hauptentität groß in der Mitte dargestellt wird. Durch die Gruppierung nach Komponenten und die Verwendung des Standard-Layouts wird das Diagramm äußerst groß (Tapete). Dies wird in Kauf genommen: Das Layout ist mit die wichtigste Information. Die Optimierung des Diagramms in Hinblick auf den benötigten Platz ist nicht gewünscht.*

*Beispiel für ein logisches Datenmodell*

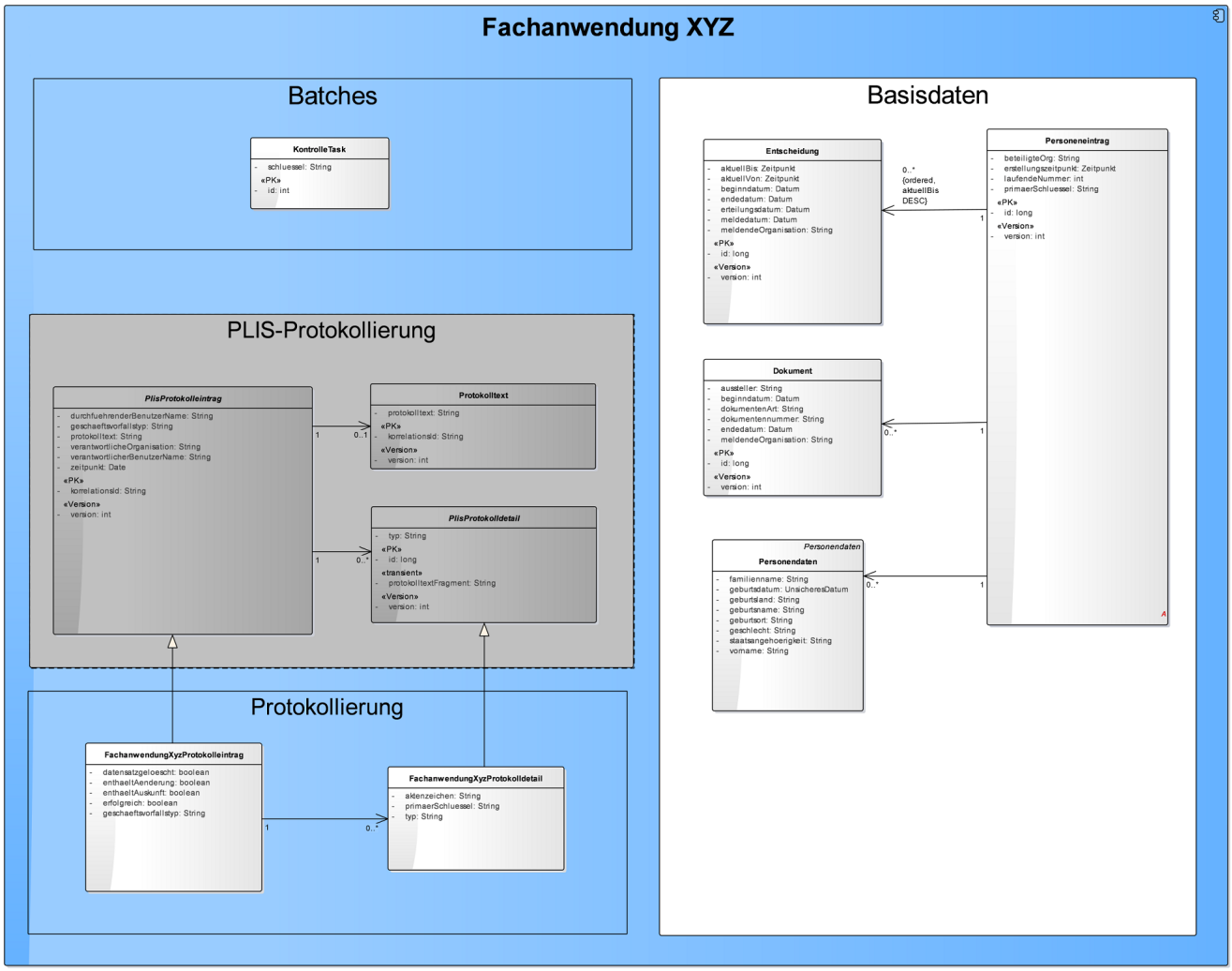


*Abbildung 8. Logisches Datenmodell*

**Physisches Datenmodell**

Das physische Datenmodell ist die Umsetzung des logischen Datenmodells in der DB. Dort wird z. B. festgelegt, an welchen Stellen Denormalisierungen und Redundanzen aufgenommen werden oder wie Vererbungsbeziehungen abgebildet werden.

Zum physischen Datenmodell werden ebenfalls ein oder mehrere UML-Diagramme analog zum logischen Datenmodell erzeugt.

***Darstellung***

- *Dargestellt werden Tabellen über Class-Elemente mit Stereotyp table.*
- *Die Komponenten, zu denen eine Gruppe von Tabellen gehört, werden über Boundary-Elemente dargestellt.*
- *Die Beziehungen zwischen Tabellen werden über Assoziation-Verbinder dargestellt.*

***Layout***

*Die Tabellen sollen nach Komponenten gruppiert werden. Jede Komponente soll als Boundary um die Tabellen der Komponente dargestellt werden. Innerhalb einer Komponente sollen die folgenden Regeln für Tabellen eingehalten werden:*

- *Voneinander erbende Tabellen sollen untereinander (von oben nach unten) dargestellt werden.*
- *Dekomposition von Tabellen soll von links nach rechts dargestellt werden: Bestandteil-Tabellen stehen rechts von ihren Haupt-Tabellen.*

*Die Wichtigkeit von Tabellen soll über ihre Position und Größe dargestellt werden. Das Ziel ist ein intuitiv möglichst gut verständliches Diagramm zu erstellen, in dem z.B. die Hauptentität groß in der Mitte dargestellt wird. Durch die Gruppierung nach Komponenten und die Verwendung des Standard-Layouts wird das Diagramm äußerst groß (Tapete). Dies wird in Kauf genommen: Das Layout ist mit die wichtigste Information. Die Optimierung des Diagramms in Hinblick auf den benötigten Platz ist nicht gewünscht.*

# Systemarchitektur des Teilsystems B

*Analog zu oben, für jedes Teilsystem entsteht ein eigenes Unterkapitel 3.x.*

# Systemarchitektur des Teilsystems C

*Analog zu oben, für jedes Teilsystem entsteht ein eigenes Unterkapitel 3.x.*

# Querschnittskonzepte

*In der Regel sind für alle wichtigen Querschnittskonzepte wie z. B. Fehlerbehandlung, Logging, Konfiguration oder Authentifizierung IsyFact-Architekturdokumente vorhanden, deshalb müssen diese hier nicht wieder aufgeführt werden.*

*In diesem Abschnitt genannt werden können:*

- *Verfeinerungen oder Konkretisierungen zu bestehenden Konzepten, z. B.:*
  - *Konfiguration: die konkrete Ausgestaltung der Konfiguration*
  - *Fehlerbehandlung: konkrete Fehlerklassen*
  - *Regelwerk: konkrete Hilfsklassen zum Regelwerk*
  - *Authentifizierung und Autorisierung: Rollen und Rechte*
- *Querschnittskonzepte, zu denen noch kein IsyFact-Standard existiert.*

# Systementwicklung

*Verweis auf das Entwicklerhandbuch. Hier kann entweder ein eigenes Entwicklerhandbuch erstellt werden und ein Verweis hierauf in das Gesamtentwicklerhandbuch aufgenommen werden, oder der Inhalt kann direkt in das Gesamtentwicklerhandbuch aufgenommen werden.*

*Die Tiefe dieses Kapitel sollte den Vorkenntnissen des Entwicklerteams angepasst werden. Es ist kein Selbstzweck und kann ggf. sogar ganz entfallen bzw. nur auf sehr wenige Punkte beschränkt sein.*

*Zu folgenden Themen sollen dem Entwicklerteam die Vorgaben klar sein:*

- *Konventionen (Namenskonventionen, Packagestruktur, ...)*
- *Entwicklungsumgebung (IDE, Tools, ...)*
- *Codegenerierung (was wird wie generiert)*
- *Konfigurationsmanagement (welche Tools, welche Verfahren, ...)*
- *Buildmanagement (welche Tools, welche Verfahren, ...)*

# Betrieb

*Verweis auf das Betriebshandbuch.*

*Auf folgende Themen ist zu referenzieren:*

- *Migration (Datenmigration, ...)*
- *Schulung*
- *Installation (Systemübergabe (Deployment), Konfiguration, ...)*
- *Systemüberwachung (Logging, Monitoring, ...)*

# Anhang

## Anhang A: Optionaler Inhalt

*Im Anhang können optionale Inhalte aufgeführt werden.*

*Sämtliche relevanten Dokumente der IsyFact werden in der Tabelle 1 aufgeführt. Die Referenzen auf diese Dokumente sind im automatisch generiertem Kapitel Literaturverweise aufgeführt.*

### *Hinweis zur Dokumentenerstellung*

Automatisch generierte Kapitel:

*Die Kapitel*

- *Literaturverweise*
- *Tabellenverzeichnis*
- *Abbildungsverzeichnis*

*werden automatisch generiert und am Ende des Dokuments eingefügt.*

*Diese Kapitel sind daher* ***nicht manuell anzulegen!***