<Beispielsystem>

Systementwurf

Version IsyFact 5.x, 03.11.2026

Inhaltsverzeichnis

Allgemeine Hinweise zur Dokumentvorlage . . . 1
- Zweck . . . 1
- Umfang . . . 1
- Nutzung . . . 1
- Bezug zum V-Modell® XT . . . 2

Einleitung . . . 3
- Zusammenfassung . . . 3
- Ziel des Dokuments . . . 3
- Zielgruppen . . . 3

Fachlicher Überblick . . . 4
- Abgrenzung (Scope) . . . 4
- Einbettung in die Anwendungslandschaft . . . 4
- Fachliche Randbedingungen und Annahmen . . . 5
- Funktionale Anforderungen . . . 5
- Nichtfunktionale Anforderungen . . . 5

SW-Architektur BSP-GA . . . 6
- Systemüberblick . . . 6
- Technische Randbedingungen und Annahmen . . . 7
- Nichtfunktionale Anforderungen . . . 7
- Konformität zur Referenzarchitektur . . . 8
- Abweichungen und Erweiterungen zur Referenzarchitektur . . . 10
- TI-Architektur . . . 11
- Querschnittskonzepte . . . 13

SW-Architektur Backend BSP_1+2-GK . . . 14
- Überblick . . . 14
- Bibliotheken & Produkte . . . 15
- Anwendungskern . . . 16
- Persistenzschicht . . . 17
- Serviceschicht . . . 17
- Querschnitt . . . 17
- Datenmodell . . . 18

SW-Architektur Batch BSP_1+2-GKBATCH . . . 21
- Überblick . . . 21
- Bibliotheken & Produkte . . . 21
- Batchschicht . . . 22

SW-Architektur <IT-Systemtyp> <IT-System-Bezeichner> . . . 23

Systementwicklung . . . 24

Betrieb . . . 25

Anhang 26
Anhang A: Optionaler Inhalt 26

Allgemeine Hinweise zur Dokumentvorlage



Dieses Kapitel beschreibt den Zweck der Vorlage zum Systementwurf und gibt allgemeine Hinweise zum Ausfüllen.

Es ist vor der Veröffentlichung des Systementwurfs zu entfernen!

Zweck

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

Der Systementwurf dient dabei mehreren Zwecken:

Grundlage für die Abstimmung mit dem Auftraggeber

Verschiedene Aspekte, wie zum Beispiel die Einbettung in die Anwendungslandschaft, müssen frühzeitig abgestimmt werden.

Anleitung der Entwicklung, Vorbereitung auf die Implementierung

Je nach Vorkenntnissen der SW-Entwickler und der Komplexität den umzusetzenden fachlichen Anforderungen müssen diese Vorgaben angemessen detailliert sein.

Umfang

In der Regel wird für jede Anwendung jeweils ein eigener Systementwurf angelegt. Für einzelne IT-Systeme der Anwendung kann ein eigener Systementwurf angelegt werden, falls für die IT-Systeme grundsätzlich andere Vorgaben oder Architekturentscheidungen vorliegen, und ein gemeinsamer Systementwurf daher zu unübersichtlich wäre.

Nutzung

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

Beschreibung der Software-Architektur

Der Systementwurf folgt dem Ansatz "top-down":

- Zunächst zeigt er, wie sich die Anwendung in die Anwendungslandschaft einfügt.
  Kapitel: Einbettung in die Anwendungslandschaft
- Danach stellt er die Zerlegung der Anwendung in IT-Systeme dar und beschreibt, wie die IT-Systeme zusammenarbeiten.
  Kapitel: Systemüberblick
- Danach beschreibt er die IT-Systeme. Zunächst werden die Komponenten des IT-Systems (Schichten bzw. Fachkomponenten) und ihr Zusammenspiel erklärt, dann wieder die

Fachkomponente 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 an anderer Stelle, insbesondere in der Referenzarchitektur, den Bausteinen und der Methodik der IsyFact, enthalten ist.

Werkzeugunterstützung bei der Erstellung von UML-Modellen

Für die Erstellung der UML-Modelle, die in diesem Dokument als Beispiele aufgeführt sind, sind die Empfehlungen der IsyFact zu beachten. Die IsyFact empfiehlt die Werkzeuge "diagrams.net/draw.io" sowie "Enterprise Architect".

Bezug zum V-Modell® XT

Der Bezug zum V-Modell® XT ist in der Methodik der IsyFact beschrieben.

Einleitung



Der Systementwurf enthält alle Informationen, welche für die an der Erstellung der Software beteiligten Parteien (Betrieb, Netzwerk-Administration, Auftraggeber, Entwickler) benötigt werden. Er 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.

Systementwürfe sollen grundsätzlich mit der vorliegenden Vorlage und im Einklang zu den beschriebenen Vorgaben und Ausfüllhinweisen erstellt werden.

Zusammenfassung



Dieser Abschnitt beinhaltet eine knappe Zusammenfassung der Anwendung:

- Was ist ihr Hintergrund?
- Wozu dient sie?

Der Abschnitt kann auf die Zusammenfassung der Systemspezifikation verweisen.

Ziel des Dokuments

Der Systementwurf dient folgenden Zwecken:

- Er ist die Grundlage zur Abstimmung zwischen dem Projektmanagement, der Softwareentwicklung des Auftraggebers/–nehmers und dem Systembetrieb.
- Er dient als Vorgabe und zur Anleitung für die Entwicklung.
- Er ist Grundlage für die Erstellung und frühzeitige Abstimmung weiterer Dokumente (zum Beispiel des Sicherheitskonzepts nach IT-Grundschutz).

Zielgruppen



Dieser Abschnitt beinhaltet eine Auflistung derjenigen, an die sich der Systementwurf richtet: Entwicklung, Betrieb, …

Für welche Zielgruppe sind welche Kapitel hauptsächlich interessant?

Fachlicher Überblick

Abgrenzung (Scope)



Knapper Text darüber, was die Aufgaben der Anwendung sind, und was nicht.

Einbettung in die Anwendungslandschaft



Dieser Abschnitt beschreibt, wie sich die zu entwickelnde Anwendung (im Beispiel: BSP-GA) in die Anwendungslandschaft einfügt. Dies wird in der Regel durch ein UML-Diagramm der Anwendungslandschaft dargestellt, in der die Anwendung hervorgehoben ist.



Das UML-Diagramm der Anwendungslandschaft sollte so abstrahiert sein, dass es nicht oft Änderungen von außen unterworfen ist. Ein solches Diagramm veraltet sehr schnell und macht häufige Änderungen am Entwurf nötig, ohne dass die Anwendung selbst sich ändert.

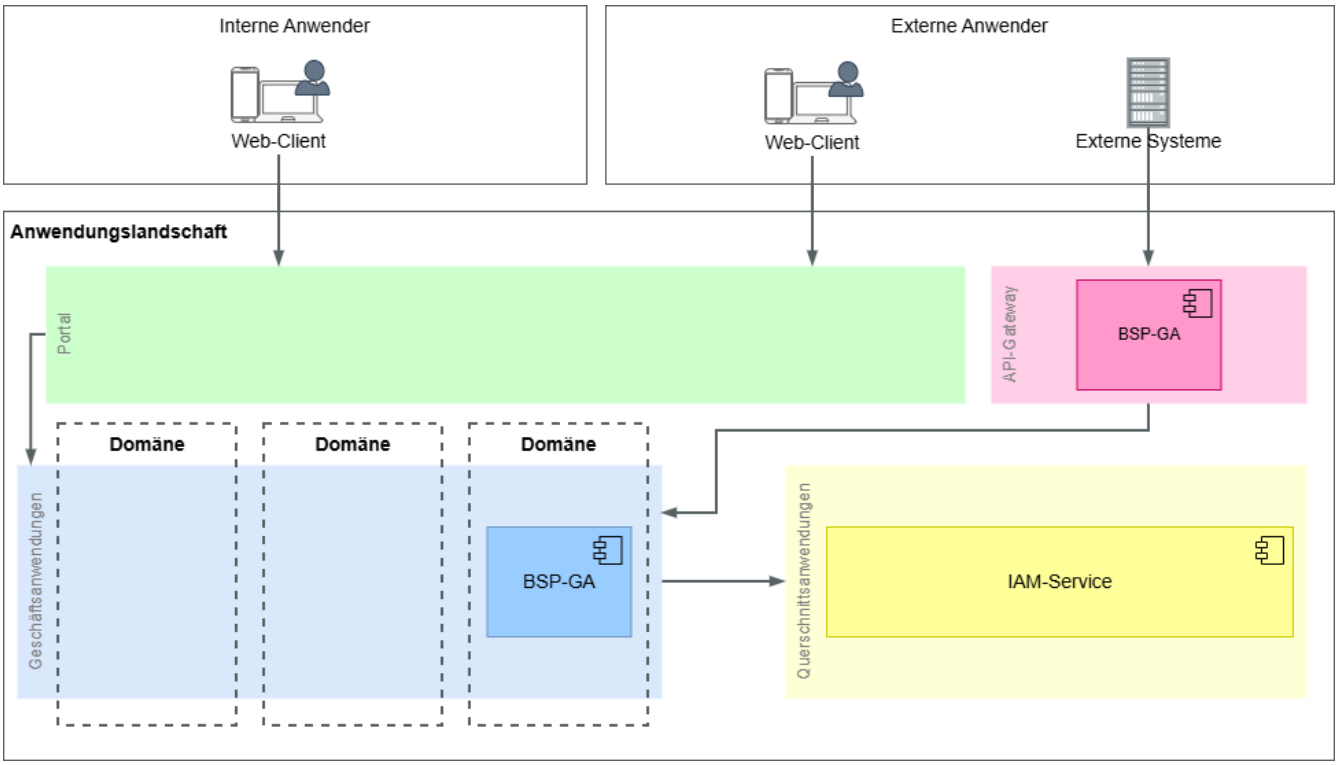


Abbildung 1. Einbettung der BSP-GA in die Anwendungslandschaft



Anwendungen, mit denen die zu entwickelnde Anwendung (im Beispiel: BSP-GA) nicht kommuniziert, sollten nicht dargestellt werden. Die Domänen, die sich selten ändern, können hingegen umrissen werden.

Fachliche Randbedingungen und Annahmen



Dieser Abschnitt verweist auf die Randbedingungen und Annahmen in der Systemspezifikation.

Fachliche Randbedingungen und Annahmen sind in der Systemspezifikation erreichbar unter (Verweis/URL) beschrieben.

Funktionale Anforderungen



Referenz auf die Quelle der Anforderungen (Anforderungsliste, Systemspezifikation, Ticketsystem, ...)

Die funktionalen Anforderungen sind in der (Quelle) erreichbar unter (Verweis/URL) beschrieben.

Nichtfunktionale Anforderungen



Referenz auf die Quelle der Anforderungen (Anforderungsliste, Systemspezifikation, Ticketsystem, ...)

Die nichtfunktionalen Anforderungen sind in der (Quelle) erreichbar unter (Verweis/URL) beschrieben.

SW-Architektur BSP-GA

Dieses Kapitel stellt den Überblick über die technische Umsetzung der Geschäftsanwendung dar. Es beschreibt außerdem die Aspekte, die für mehrere der IT-Systeme gelten, in welche die Geschäftsanwendung zerlegt wird. Das können Verfahren, Festlegungen oder die Beschreibung gemeinsam genutzter Komponenten sein. In der Regel zerlegt sich eine Geschäftsanwendung in:



- mindestens ein Backend,
- ein Frontend (beim Vorhandensein einer grafischen Oberfläche),
- Batches,
- Gateways, falls externe Zugriffe nötig sind.

Festlegungen, die bereits in übergeordneten Dokumenten der IsyFact oder der Organisation (z.B. IsyFact-Referenzarchitektur, Sicherheitsrichtlinie der Organisation, ...) enthalten sind, werden referenziert. **Abweichungen hiervon sind zu begründen!**

Systemüberblick

Der Systemüberblick beschreibt, in welche IT-Systeme die Geschäftsanwendung zerlegt wird, und von welchem IT-Systemtyp diese sind. Die Referenzarchitektur der IsyFact definiert die IT-Systemtypen Backend, Frontend, Batch und Gateway. Dieses Kapitel ist die nächste Stufe der Konkretisierung nach der Einbettung in die Anwendungslandschaft.

Das Kapitel besteht aus einem UML-Diagramm, das darstellt, aus welchen IT-Systemen die Geschäftsanwendung besteht, wie diese sich gegenseitig aufrufen und welche Nachbarsysteme 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. Übersichtlichkeit steht klar über der Beschreibung von Details!



Das UML-Diagramm beschreibt ein begleitender Text, der sich folgender Fragen annimmt:

- Was sind die Aufgaben der einzelnen IT-Systeme? Welche Funktionalität bieten sie an?
- Wie sieht das Zusammenspiel der IT-Systeme aus, um die Gesamtaufgabe des Systems zu erfüllen? Wie sieht die Aufgabenverteilung aus?

Die folgende Abbildung 2 basiert auf der Beispielzerlegung der Referenzarchitektur. Es zeigt nicht nur die Zerlegung, sondern schafft auch den Rückbezug auf die Systemspezifikation. Wenn dies der Übersicht nicht dienlich ist, kann der Rückbezug auch textuell oder tabellarisch hergestellt werden.

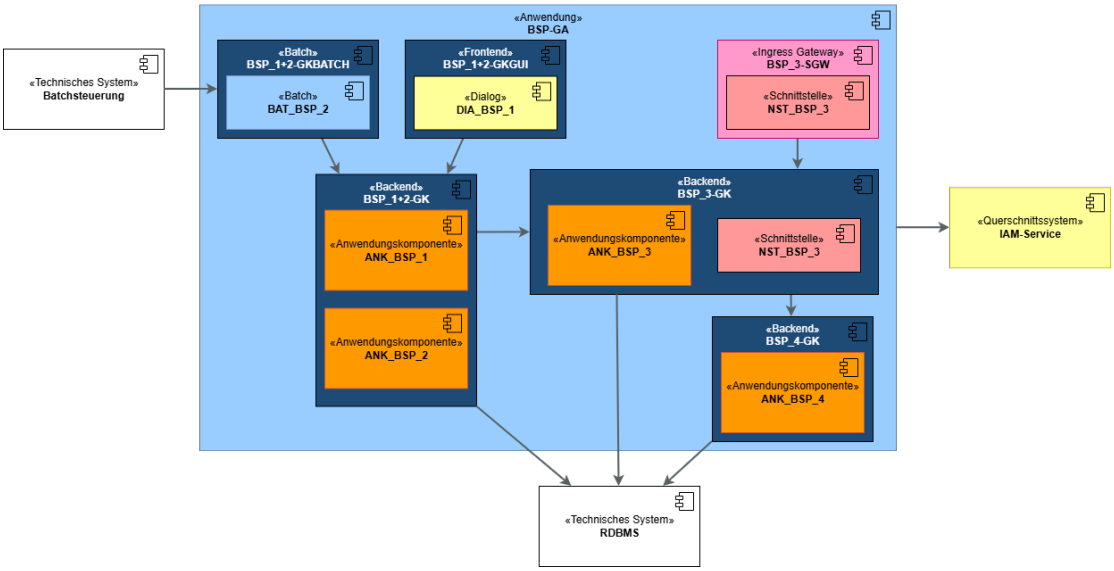


Abbildung 2. Zerlegung der BSP-GA in IT-Systeme

Im begleitenden Text wären die Aufrufbeziehungen der IT-Systeme untereinander knapp zu erläutern. Ebenso wäre zu beschreiben, wieso die Verbindung zum IAM-Service von der fachlichen Klammer der Anwendung ausgeht. In diesem Beispiel soll dies verdeutlichen, dass alle enthaltenen IT-Systeme den IAM-Service aufrufen.

Technische Randbedingungen und Annahmen

Dieser Abschnitt beschreibt technische Randbedingungen und Annahmen, die für den Systementwurf relevant sind, ausführlich.

Annahmen sind Aussagen, deren Gültigkeit für möglich gehalten wird, die aber nicht bewiesen oder verifiziert sind. Annahmen benötigen immer eine Begründung (das heißt: "Warum nehmen wir das an?") und sollten während des Vorhabens periodisch überprüft werden.

Randbedingungen (auch als Rahmenbedingungen bezeichnet) sind Umstände, die nur mit großem Aufwand oder gar nicht beeinflussbar sind oder sich aus der Problemstellung zwingend ergeben, und daher als gegeben betrachtet werden.

Nichtfunktionale Anforderungen

Architekturrelevante nichtfunktionale Anforderungen konkretisieren in der Regel die Qualitätsziele der Anwendung. Sie machen diese Qualitätsziele messbar und durch Tests überprüfbar.

Die Strukturierung dieses Kapitels sollte sich an den Qualitätskriterien der ISO 25010 orientieren:

- Funktionale Eignung,
- Sicherheit,

- Effizienz,
- Zuverlässigkeit,
- Benutzbarkeit,
- Wartbarkeit,
- Kompatibilität,
- Übertragbarkeit.

Konformität zur Referenzarchitektur

In diesem Kapitel wird die Konformität des Systems zur Referenzarchitektur bestätigt und auf eventuell vorhandene Erweiterungen, Abweichungen und abweichende Versionen verwiesen. Eine Übersicht über die Software-Artefakte der IsyFact findet sich in Tabelle 1.

Die Spalte "Status" nimmt die folgenden Werte an:

berücksichtigt

Das Element wird in der Standard-Version des aktuellen IsyFact-Releases berücksichtigt.

nicht benötigt

Das Element ist nicht relevant.

wird erweitert

Das Element wird erweitert, um den Anforderungen des Gesamtsystems gerecht zu werden.

abweichend

Das Element wird mit Abweichungen berücksichtigt, um den Anforderungen des Systems gerecht zu werden.

abweichende Version

Das Element wird in einer abweichenden Version zu der des aktuellen IsyFact-Releases berücksichtigt. Die Spalte "Version" enthält die abweichende Version des Elements.

In den letzten drei Fällen (Erweiterung, Abweichung sowie abweichende Versionen) sind im folgenden Kapitel, Abweichungen und Erweiterungen zur Referenzarchitektur, Architekturentscheidungen in Form von ADRs zu dokumentieren und in Form von Verweisen auf die ADRs in der Spalte "Referenzen" der jeweiligen Elemente aufzuführen.

Das Anwendung BSP-GA wird konform zur Referenzarchitektur der IsyFact umgesetzt. Die folgende Tabelle führt die verwendeten Elemente der IsyFact auf.

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

Thema	Status	Version	Referenzen
Allgemein			
IsyFact	berücksichtigt	5.x	
Referenzarchitektur			
Backend	berücksichtigt	*Standard*	
Batch	berücksichtigt	*Standard*	
Frontend	berücksichtigt	*Standard*	
Gateway	berücksichtigt	*Standard*	
Kommunikation	berücksichtigt	*Standard*	
Bausteine (IsyFact-Standards)			
Datum & Zeit	berücksichtigt	…	
Fehlerbehandlung	berücksichtigt	*Standard*	
Logging	berücksichtigt	*Standard*	
Polling	berücksichtigt	…	
Security	berücksichtigt	…	
Sonderzeichen	berücksichtigt	…	
Task Scheduling	Abweichungen	…	ADR-BSP-GA-1
Überwachung	berücksichtigt	*Standard*	
Util	berücksichtigt	…	
Bausteine (IsyFact-Erweiterungen)			
Angular	berücksichtigt	…	
Bedienkonzept	berücksichtigt	…	
…	…	…	
Plattform			
Deployment-Konzept	berücksichtigt	*Standard*	
Methodik			
Namenskonventionen	berücksichtigt	*Standard*	
Versionierung	berücksichtigt	*Standard*	
Java-Programmierkonventionen	berücksichtigt	*Standard*	
TypeScript-Programmierkonventionen	berücksichtigt	*Standard*	
Vorlagen	berücksichtigt	*Standard*	
Diagrammerstellung	berücksichtigt	*Standard*	
Werkzeuge			
OpenAPI Editor	berücksichtigt	*Standard*	

Thema	Status	Version	Referenzen
OpenAPI Generator	berücksichtigt	*Standard*	

Abweichungen und Erweiterungen zur Referenzarchitektur



In diesem Kapitel werden Abweichungen und Erweiterungen zur Referenzarchitektur in Form von Architecture Decision Records (ADRs) dokumentiert.

Falls mehr als die von der Vorlage vorgesehenen Informationen erfasst werden müssen, kann die ADR-Vorlage um weitere Zeilen ergänzt werden.

Es sollten auch abgelehnte und abgelöste ADRs im Systementwurf enthalten sein, um das in ihnen enthaltene Wissen festzuhalten.

Die folgenden Architekturentscheidungen begründen Abweichungen und Erweiterungen zur Referenzarchitektur. Sie gelten für das im Dokument beschriebene Gesamtsystem. Darüber hinaus sind folgende, übergreifende Architekturentscheidungen zu beachten: übergreifende Architekturentscheidungen.

ADR-BSP-GA-1: Titel

Entscheidung	*Architekturentscheidung im Wortlaut.* ***Empfohlener Umfang:*** *wenige Sätze.*
Begründung	*Begründung der Architekturentscheidung.* ***Empfohlener Umfang:*** *wesentliche Argumente, die zur Architekturentscheidung geführt haben, in jeweils wenigen Sätzen.* ***Optional:*** *wesentliche Gegenargumente, die von den Argumenten überwogen wurden, in jeweils wenigen Sätzen.*
Entscheidende	*Aufzählung der Personen, welche die Entscheidung getroffen haben. Im Fall einer Abstimmung auch Enthaltungen und Gegenstimmen dokumentieren, ggf. mit kurzem Bezug auf wesentliche Gegenargumente.* ***Falls vorhanden:*** *Verweis auf das Protokoll, in dem der Beschluss dokumentiert ist.*
Datum	*Datum der Entscheidung.*
Alternativen	***Optional:*** *Beschreibung der Alternativen mit den wesentlichen Beweggründen ihrer Betrachtung, in jeweils wenigen Sätzen.*
Auswirkungen	***Optional:*** *Beschreibung von Auswirkungen, vor allem falls nicht direkt ersichtlich oder durch Zusammenspiel mit anderen (übergreifenden) ADRs, in jeweils wenigen Sätzen.*

ADR-BSP-GA-2: Titel

...

TI-Architektur

Das Kapitel enthält eine kurze Beschreibung der wichtigsten Eckpunkte der TI-Architektur.

Die Beschreibung besteht aus Deployment-Diagrammen mit begleitendem Text. Die Diagramme beschreiben die Zuordnung der Server bzw. Laufzeitumgebungen zu den Zonen sowie die Zuordnung der Deployment-Einheiten auf die Server/Laufzeitumgebungen. Es enthält keine konkreten Adressen, Portnummern und Dimensionierungen, sondern eine logische Sicht.

Die Kommunikationsbeziehungen werden zum Load Balancer bzw. zur Firewall dargestellt. Es soll klar werden, über welche Zonengrenzen hinweg mit welchen Protokollen kommuniziert wird.

Existierende Nachbarsysteme werden in der Regel nicht dargestellt.

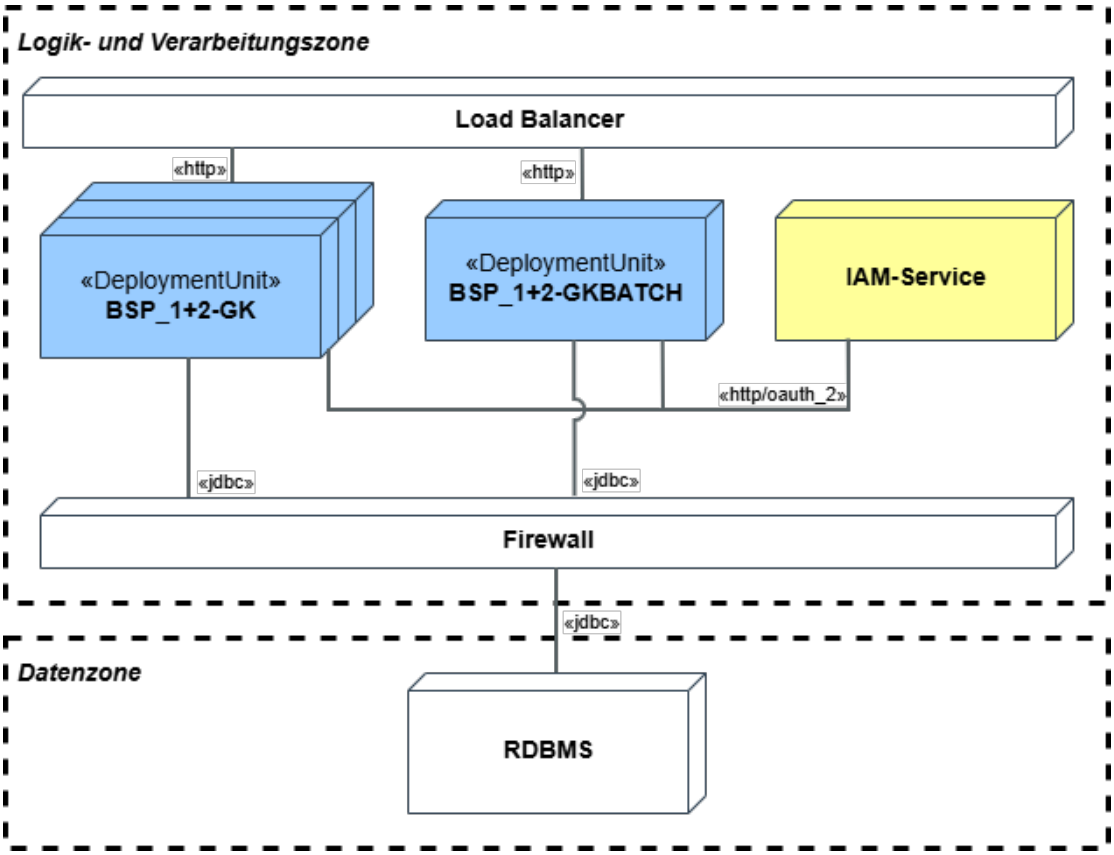


Abbildung 3. TI-Architektur BSP-GA (teilweise)

Das Beispiel-Diagramm in Abbildung 3 zeigt mit einem Backend und einem Batch einen Teil der BSP-GA.

Laufzeitumgebung

Der Abschnitt erläutert die eingesetzten Produkte für die Laufzeitumgebung. Diese Umgebung ist durch den Produktkatalog der IsyFact zum größten Teil vorgegeben.

Die Laufzeitumgebung für die IT-Systeme der Geschäftsanwendung BSP-GA besteht gemäß aus den

folgenden Produkten:

Tabelle 2. Laufzeitumgebung für IT-Systeme der BSP-GA

Kategorie	Name	Version	Bemerkung
Betriebssystem	Linux 64 bit	Kernel >= 4.12	
Java-Laufzeitumgebung	OpenJDK (Eclipse Temurin von Adoptium)	25.x	
Servlet-Container	Apache Tomcat	Z.B. 8.5.x	

Ressourcen



Der Abschnitt enthält eine Abschätzung der Ressourcen in der Produktionsumgebung:

- Mengengerüst für die Datenbank,
- Platzbedarf von (temporären) Dateien.

Die Abschätzung geschieht in tabellarischer Darstellung und mithilfe erläuterndem Text.

Mengengerüst für die Datenbank

In Tabelle 3 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 3. Mengengerüst für die Datenbank

Tabelle/Index	Typ	Anzahl/Zeilen	Größe Rohdaten		Bemerkung
			Zeile	Tabelle	
…					
…					
…					

Platzbedarf für Dateien



Der Abschnitt enthält eine Abschätzung der Ressourcen in der Produktionsumgebung:

- Mengengerüst für Dateien auf dem lokalen Laufwerk.

Die Abschätzung geschieht in tabellarischer Darstellung und mithilfe

erläuterndem Text.

In Tabelle 4 ist eine Abschätzung für die auf dem lokalen Laufwerk gespeicherten Dateien enthalten.

Tabelle 4. Mengengerüst für die Dateien auf dem lokalen Laufwerk

Datei	Größe	Bemerkung
…		
…		

Querschnittskonzepte



Dieser Abschnitt kann folgende Inhalte enthalten:

- Verfeinerungen oder Konkretisierungen zu bestehenden IsyFact-Bausteinen,
- für alle IT-Systeme geltende Abweichungen von Bausteinen,
- Querschnittskonzepte, zu denen noch kein Baustein existiert.

SW-Architektur Backend BSP_1+2-GK



Im Kapitel SW-Architektur BSP-GA wurde die in der Systemspezifikation beschriebene Anwendung in IT-Systeme zerlegt. Diese werden ab hier in eigenen Unterkapiteln der Reihe nach beschrieben.

Dieses Kapitel enthält vornehmlich eine Top-Down-Beschreibung des IT-Systems anhand seines IT-Systemtyps. Die Beschreibung umfasst die Aufteilung des IT-Systems in Komponenten und deren Bezug zur Systemspezifikation. Der Detailgrad der Beschreibung der Komponenten hängt u.a. vom Maß an nötigen Abweichungen und Erweiterungen zur Referenzarchitektur sowie von der vorhandenen Erfahrung in den Architektur- und Entwicklungsteams mit der IsyFact und den verwendeten Produkten ab.

Weiterhin referenziert das Kapitel übergeordnete Designprinzipien und Patterns, die beim Entwurf des IT-Systems eine Rolle spielen. Schließlich enthält es weitergehende Details, Hinweise und Besonderheiten, deren Kenntnis für die Entwicklung und Wartung des IT-Systems hilfreich sind.



Wenn sich Beschreibungen in den folgenden Kapiteln doppeln, ist es sinnvoll, ein Kapitel voranzustellen, das Gemeinsamkeiten aller IT-Systeme bzw. einzelner IT-Systemtypen (zum Beispiel aller Backends) erläutert. Wenn diese Gemeinsamkeiten die Referenzarchitektur oder bestehende Bausteine der IsyFact betreffen, können sie auch Einzug in die Querschnittskonzepte halten.

Dieses Kapitel beschreibt die Systemarchitektur des Backends BSP_1+2-GK.

Überblick

Dieser Abschnitt zeigt einen Überblick über das IT-System. Dies erfolgt in der Regel mit einem UML-Komponentendiagramm und einem knappen begleitenden Text, der die einzelnen Komponenten und ihr Zusammenspiel erklärt. Als Leitfaden für das Diagramm kann die Beschreibung des technischen Kontextes von arc42 dienen. Im Unterschied dazu ist hier vorgesehen, auch die Innensicht des betrachteten IT-Systems zu zeigen. Ein Diagramm, das den technischen Kontext des Gesamtsystems zeigt, befindet sich in der SW-Architektur BSP-GA.



Das Komponentendiagramm zeigt eine Übersicht der fachlichen Schnittstellen und sowohl fachlichen als auch technischen Komponenten des IT-Systems. Es stellt die Zusammenhänge und Beziehungen zwischen den Bestandteilen des IT-Systems dar. Es beschreibt nicht die Logik der Verarbeitung.

Die folgende Abbildung 4 zeigt eines der Backends aus der Beispielzerlegung der Referenzarchitektur.

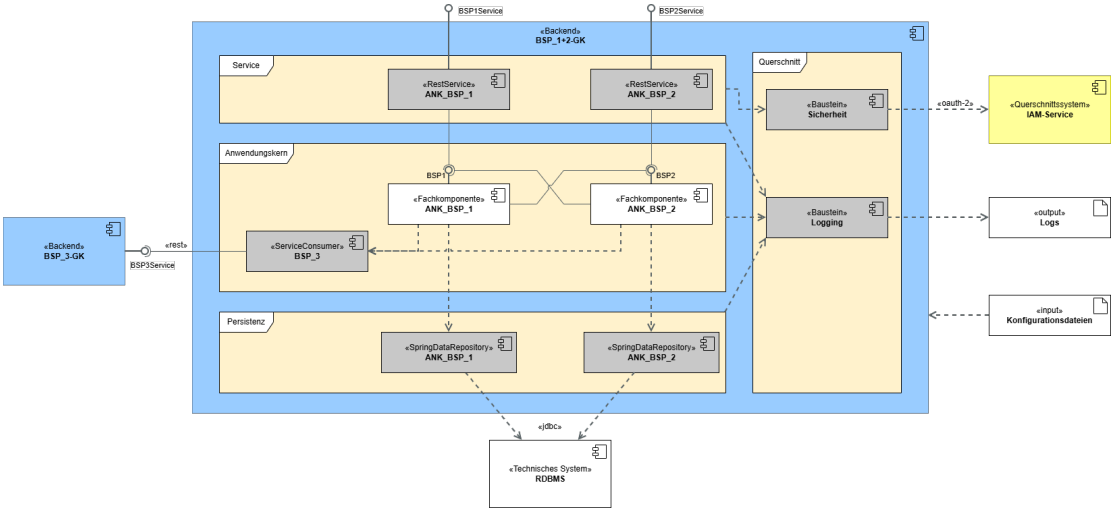


Abbildung 4. Backend der Beispielzerlegung

Hinweise zur Darstellung:

- Der Einfachheit halber sind nur fachliche Schnittstellen und diejenigen technischen Schnittstellen, die zur Kommunikation mit anderen Systemen notwendig sind, explizit dargestellt.
- UML-Komponenten außerhalb des IT-Systems stellen Systeme dar, mit denen das IT-System kommuniziert.
- Rechts neben dem IT-System sind Artefakte dargestellt, die als Ein- und Ausgabe für das IT-System gelten. Hierfür verwendet das Diagramm die Stereotypen `<<input>>` respektive `<<output>>`.
- Im Querschnitt befinden sich hauptsächlich von der IsyFact bereitgestellte Bausteine.
- Die Stereotypen der Komponenten verweisen auf die Referenzarchitektur und geben hierdurch die Umsetzung vor. In diesem Beispiel verweisen die Stereotypen auf Inhalte der Referenzarchitektur Backend.
- Das Diagramm verzichtet, vor allem innerhalb des IT-Systems, auf Stereotypen an Aufrufbeziehungen, wenn diese keine Zusatzinformation beitragen (u.a.`<<use>>`).

Bibliotheken & Produkte

Auflistung der benötigten Bibliotheken für die Entwicklung des IT-Systems in tabellarischer Darstellung.



Wenn keine Version eingetragen wird, richtet sich die Version nach dem Produktkatalog der IsyFact-Version, die in Konformität zur Referenzarchitektur benannt ist.

Abweichungen von der Referenzarchitektur sind in der Spalte "Bemerkung" zu hinterlegen. Zusätzliche, nicht von der IsyFact vorgeschriebene und empfohlene Produkte sind ebenfalls zu kennzeichnen. Transitive Abhängigkeiten sind nicht

einzutragen.

Das IT-System BSP_1+2-GK benötigt gemäß Produktkatalog der IsyFact die folgenden Bibliotheken bzw. Produkte:

Tabelle 5. BSP_1+2-GK: Bibliotheken & Produkte

Kategorie	Name	Version	Bemerkung
Programmiersprache	Java		
Application Framework	Spring Boot		
…	…		

Anwendungskern



Dieser Abschnitt beschreibt die technische Umsetzung der Anwendungskomponenten (ANK) in Fachkomponenten. Die Tiefe der Darstellung richtet sich nach der fachlichen Komplexität der umzusetzenden Funktionalität, das heißt Anwendungsfälle (AWF) und Modellkomponenten (MKO), aus der Systemspezifikation.

Die folgenden Abschnitte beschreiben die technische Umsetzung der Anwendungskomponenten.

ANK_BSP_1



Die Beschreibung einer Anwendungskomponente enthält in der Regel ein UML-Komponentendiagramm und erklärenden Text.

Das Diagramm beschreibt den Aufbau der Anwendungskomponente und gibt einen Überblick über ihre wichtigsten Klassen. Ein Beispiel für eine generische Fachkomponente mit Klassen für Anwendungsfälle und Anwendungsfunktionen ist in der Beschreibung des Anwendungskerns der Referenzarchitektur enthalten.



Rückbezug auf die Systemspezifikation

Die Komponente stellt in ihrer Beschreibung den Bezug zur Spezifikation dar:

- Wie bilden sich die Anwendungsfälle (AWF) der Spezifikation auf die Anwendungsfallklassen des Codes ab?
- Wie bilden sich die Anwendungsfunktionen (AFU) der Spezifikation auf die Anwendungsfunktionsklassen im Code ab?

ANK_BSP_2

…

Persistenzschicht



Die Vorgaben zur Persistenzschicht sind in der Referenzarchitektur Backend beschrieben.

Dieser Abschnitt kann Besonderheiten der Persistenzschicht erläutern, die über die Vorgaben der Referenzarchitektur hinausgehen. Ebenfalls sinnvoll ist der Verweis auf zusätzliche, fachliche Konzepte, die Auswirkungen auf die Persistenzschicht haben. Beispiele hierfür sind der Umgang mit Sperren oder Löschfristen.

Serviceschicht



Die Vorgaben zur Serviceschicht sind in der Referenzarchitektur Backend beschrieben.

Dieser Abschnitt kann Besonderheiten der Serviceschicht erläutern, die über die Vorgaben der Referenzarchitektur hinausgehen. Ebenfalls sinnvoll ist der Verweis auf zusätzliche, fachliche Konzepte, die Auswirkungen auf die Serviceschicht haben.

Die konkrete Konstruktion der einzelnen Services sind in einem eigenen Unterabschnitt pro Service beschrieben.



Rückbezug auf die Systemspezifikation

Die Komponente stellt in ihrer Beschreibung den Bezug zur Spezifikation dar:

- Wie bilden sich die Nachbarsystemschnittstellen (NST) der Spezifikation auf die Klassen des Codes ab?

BSP1Service

...

BSP2Service

...

Querschnitt



Dieser optionale Abschnitt enthält Konzepte des Querschnitts, welche nur für das vorliegende Teilsystem gelten.

Datenmodell

Beschreibung des logischen und physischen Datenmodells des Backends als UML-Klassendiagramm.

Die Entitäten des Datenmodells werden den Fachkomponenten des Anwendungskerns zugeordnet. Die Fachkomponenten werden in den Diagrammen durch Boundaries repräsentiert.

Das übliche Vorgehen besteht aus den folgenden Schritten:

1. Die Entitäten werden aus dem fachlichen Modell (Datenmodell der Systemspezifikation) in das technische Modell übernommen. 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. Das technische Modell wird in Grafiken für den Systementwurf überführt. Ist das so entstandene technische Datenmodell zu groß und unübersichtlich für eine einzelne Grafik, wird es in Teildiagramme zerlegt. Grundsätzlich orientieren sich die Teildiagramme am Schnitt der Fachkomponenten, können bei Bedarf aber auch feiner zerlegt werden. Zusätzlich sollte das gesamte Datenmodell in Form einer groben Übersicht (z. B. ohne Attribute) dargestellt werden, um den Lesenden 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 des IT-Systems. 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 Primärschlüssel 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 die Anordnung von 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

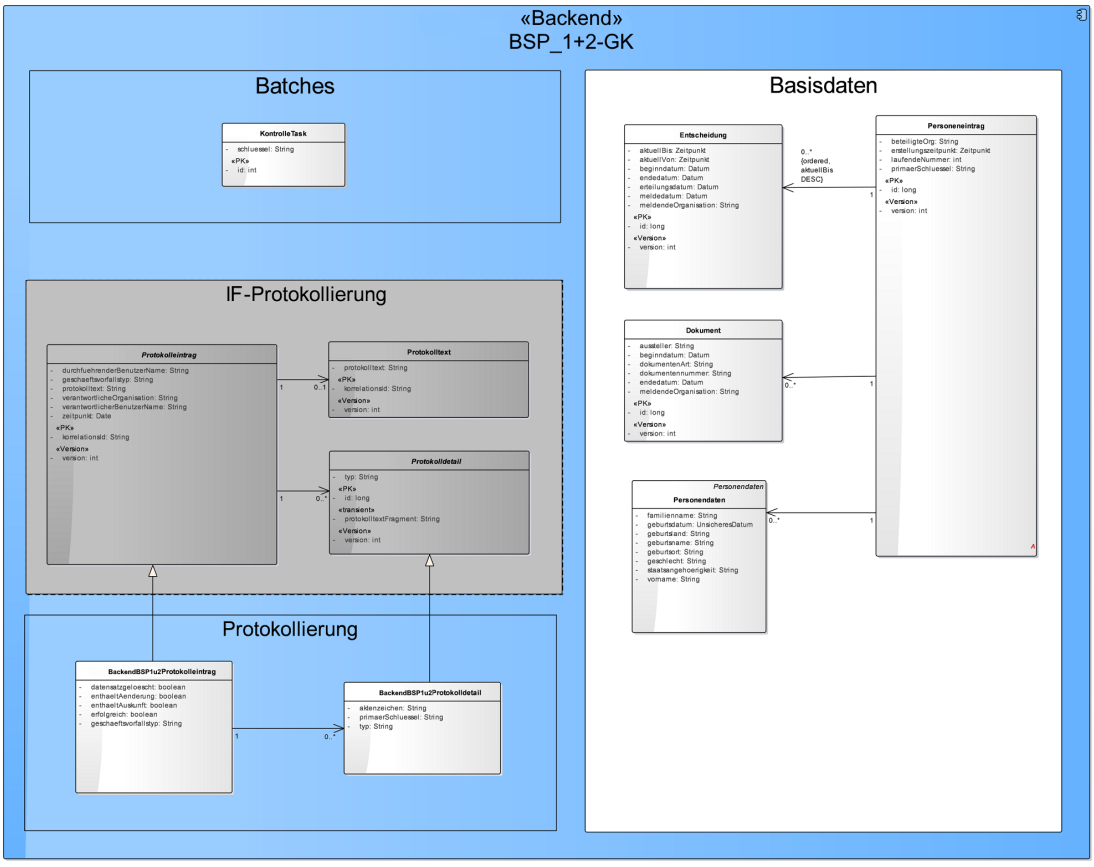


Abbildung 5. Logisches Datenmodell

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 die Anordnung von 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.

SW-Architektur Batch BSP_1+2-GKBATCH



Die allgemeinen Vorgaben zu Batches sind in der Referenzarchitektur Batch der IsyFact beschrieben. Sie können bei Bedarf referenziert werden.

Da sich Batches laut Referenzarchitektur oft den Anwendungskern und die Persistenzschicht mit einem Backend teilen, kann auf die entsprechenden Kapitel referenziert werden. Falls dies nicht der Fall ist, ist das Backend wie ein Nachbarsystem zu behandeln und die Batches erhalten einen Anwendungskern mit einem Service Consumer pro benötigtem Service.

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.

Überblick



Der Überblick richtet sich nach dem in Überblick beschriebenen Vorgehen.

Bibliotheken & Produkte



Auflistung der benötigten Bibliotheken für die Entwicklung des IT-Systems in tabellarischer Darstellung.

Wenn keine Version eingetragen wird, richtet sich die Version nach dem Produktkatalog der IsyFact-Version, die in Konformität zur Referenzarchitektur benannt ist.

Abweichungen von der Referenzarchitektur sind in der Spalte "Bemerkung" zu hinterlegen. Zusätzliche, nicht von der IsyFact vorgeschriebene und empfohlene Produkte sind ebenfalls zu kennzeichnen. Transitive Abhängigkeiten sind nicht einzutragen.

Das IT-System BSP_1+2-GKBATCH benötigt gemäß Produktkatalog der IsyFact die folgenden Bibliotheken bzw. Produkte:

Tabelle 6. BSP_1+2-GKBATCH: Bibliotheken & Produkte

Kategorie	Name	Version	Bemerkung
Programmiersprache	Java		
Application Framework	Spring Boot		
Batch-Framework	Spring Batch	s. Spring Boot Dependency Management	nur für BSP_1+2-GKBATCH, Entscheidung für Alternativprodukt gemäß ADR-BSP-GA-1

Kategorie	Name	Version	Bemerkung
…	…		

Batchschicht



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

BAT_BSP_2

…

SW-Architektur <IT-Systemtyp> <IT-System-Bezeichner>



Analog zu SW-Architektur Backend BSP_1+2-GK und SW-Architektur Batch BSP_1+2-GKBATCH: für jedes IT-System entsteht ein eigenes Unterkapitel gemäß seines IT-Systemtyps.

Systementwicklung

Dieses Kapitel verweist, wo möglich, auf übergreifende Regelungen (Entwicklungshandbuch, Coding Guidelines, …).

Die Tiefe dieses Kapitel passt sich den Vorkenntnissen des Entwicklungsteams an. Es ist kein Selbstzweck und kann sich auf wenige Punkte beschränken oder ggf. sogar ganz entfallen.



Wenn es keine übergreifenden Regelungen gibt, adressiert das Kapitel unter anderem folgende Themen:

- Konventionen (Namenskonventionen, Code-Struktur, …),
- Entwicklungsumgebung (IDEs, Tools, …),
- Codegenerierung (was und wie),
- Konfigurationsmanagement (Tools, Verfahren, …),
- Buildmanagement (Tools, Verfahren, …).

Betrieb



Dieses Kapitel verweist, wo möglich, auf ein Betriebshandbuch (oder dazu ähnliche Dokumente).

Es verweist in der Regel auf folgende Themen:

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

Anhang



Der Anhang führt weiterführende oder vertiefende Inhalte auf, die den Lesefluss und die Übersicht des Dokuments "an Ort und Stelle" stören würde. Beispiele hierzu sind Templates für Dateien, zum Beispiel zur Konfiguration oder zum Build.

Anhang A: Optionaler Inhalt

...