



Bundesverwaltungsamt



## IsyFact-Standard

# IsyFact – Einstieg

**Version 1.5**  
**12.05.2015**



„IsyFact - Einstieg“ des Bundesverwaltungsamts ist lizenziert unter einer Creative Commons Namensnennung 4.0 International Lizenz.



„IsyFact - Einstieg“  
des Bundesverwaltungsamts ist lizenziert unter einer  
Creative Commons Namensnennung 4.0 International Lizenz.

Die Lizenzbestimmungen können unter folgender URL heruntergeladen  
werden: <http://creativecommons.org/licenses/by/4.0>

**Ansprechpartner:**

Referat Z II 2  
Bundesverwaltungsamt  
E-Mail: [isyfact@bva.bund.de](mailto:isyfact@bva.bund.de)  
Internet: [www.isyfact.de](http://www.isyfact.de)

## Dokumentinformationen

Dokumenten-ID:	IsyFact-Einstieg.docx
----------------	-----------------------

# Inhaltsverzeichnis

<b>1. Einleitung .....</b>	<b>6</b>
1.1. Aufbau und Zweck des Dokuments.....	6
<b>2. Vorstellung der IsyFact .....</b>	<b>7</b>
2.1. Einführung in Software Factories .....	7
2.2. IsyFact: Standards und Erweiterungen .....	7
2.3. Nutzungsszenarien und Tailoring.....	9
2.4. Festlegungen der IsyFact-Konformität .....	10
2.5. Mitarbeit an der IsyFact.....	11
2.6. Historie der IsyFact .....	12
<b>3. Rahmenbedingungen .....</b>	<b>14</b>
3.1. Auswirkung der Historie der IsyFact.....	14
3.2. Aktueller Stand und Weiterentwicklung.....	14
3.3. Verwendete Software-Produkte .....	15
3.4. Annahmen zu Projektrollen .....	15
3.5. Styleguides und die Gestaltung von Benutzeroberflächen.....	16
<b>4. IsyFact-Standards .....</b>	<b>17</b>
4.1. Vorgaben für die Spezifikation .....	17
4.2. Vorgaben für den Systementwurf.....	18
4.2.1 Vorgaben für die A-Architektur.....	18
4.2.2 Vorgaben für die T-Architektur .....	19
4.2.2.1 Datenzugriff .....	19
4.2.2.2 Anwendungskern .....	20
4.2.2.3 Batch .....	20
4.2.2.4 Service.....	20
4.2.2.5 GUI .....	21
4.2.3 Vorgaben für die TI-Architektur .....	21
4.3. Vorgaben für die Realisierung.....	21
4.4. Querschnittlich zu nutzende Basiskomponenten .....	22
4.4.1 Fehlerbehandlung .....	22
4.4.2 Administrative Überwachung und Konfiguration .....	22
4.4.3 Behandlung von internationalen Sonderzeichen .....	23

4.4.4	Logging .....	23
4.4.5	Berechtigungen .....	23
4.4.6	LDAP-Zugriffe .....	23
4.4.7	Polling .....	24
4.4.8	Util .....	24
<b>5.</b>	<b>IsyFact-Erweiterungen .....</b>	<b>25</b>
<b>6.</b>	<b>Erste Schritte .....</b>	<b>26</b>
<b>7.</b>	<b>Abbildungsverzeichnis .....</b>	<b>27</b>

## 1. Einleitung

### 1.1. Aufbau und Zweck des Dokuments

Dieses Dokument gibt einen Überblick über die Dokumente und Vorgaben für die Entwicklung von IT-Systemen gemäß den Konzepten der IsyFact-Standards. Dabei werden die für die Entwicklung relevanten Dokumente referenziert. Es werden Grundlagen zum Aufbau von Fachanwendungen nach IsyFact-Referenzarchitektur kurz eingeführt. Detaillierte Inhalte der referenzierten Dokumente werden allerdings nicht wiederholt. Das Dokument ist wie folgt unterteilt:

Im Kapitel **Vorstellung der IsyFact** werden die grundlegenden Aspekte und die Vision der IsyFact dargestellt, deren Fundament die IsyFact-Standards bilden.

Im Kapitel **Rahmenbedingungen** werden querschnittliche Aspekte beschrieben, die beim Einsatz der IsyFact zu beachten sind.

In den darauffolgenden Kapiteln werden die Inhalte der **IsyFact-Standards** dargestellt und einen Ausblick auf die **IsyFact-Erweiterungen** geliefert.

Im Kapitel **Erste Schritte** werden die konkreten Schritte dargestellt, die zur Einführung der IsyFact in einem neuen Kontext durchzuführen sind.

## 2. Vorstellung der IsyFact

In diesem Kapitel werden die IsyFact und die zugrundeliegenden Prinzipien erläutert.

### 2.1. Einführung in Software Factories

Bei der Entwicklung von Anwendungssystemen für eine bestimmte Domäne stehen Architekten und Entwickler häufig vor denselben, wiederkehrenden Problemen. Zwar unterscheiden sich je nach Anwendung die Datenstrukturen, die Geschäftslogik und die Benutzeroberflächen, aber die allgemeinen Prinzipien der Architektur, der Datenhaltung und der Gestaltung von Benutzerschnittstellen, sind in der Regel für alle Anwendungen identisch und somit wiederverwendbar. Ebenso lässt sich eine Vielzahl von Detailvorgaben, z.B. für den Betrieb von Anwendungen, das Logging, die Benutzerverwaltung oder für die Kommunikation von Systemen untereinander standardisieren. Dies ist nicht nur sinnvoll, sondern auch notwendig, wenn die verschiedenen Anwendungssysteme Teil einer Anwendungslandschaft in einem Unternehmen oder einer Behörde sind. Erst durch die Wiederverwendung von bereits entwickelten Komponenten und die Standardisierung von Entwicklungsmethoden wird es möglich, eine komplexe Anwendungslandschaft mit geringem Aufwand bzw. kostengünstig zu entwickeln und effizient zu betreiben.

Die Herausforderung bei der Wiederverwendung besteht darin, das einmal erworbene Wissen über die Anwendungsentwicklung in einer Domäne so zu strukturieren, zu dokumentieren und vorzuhalten, dass nachfolgende Projekte einfach und verlässlich darauf zugreifen können, und damit die Einhaltung des einmal definierten Architekturrahmens sichergestellt ist. Software Factories sind ein Mittel, dies zu erreichen. Eine Software Factory ist eine Sammlung von Architektur-Blaupausen, Bausteinen, einer Plattform, Methoden und Werkzeugen, die es erlaubt, Anwendungen weitgehend „automatisiert“ und standardisiert zu entwickeln. Dabei bedeutet „automatisiert“ hier nicht, dass der Quelltext des Software Systems ohne menschliches Zutun durch einen Automaten erzeugt würde. Es bezieht sich vielmehr darauf, dass alle wesentlichen technischen Architekturentscheidungen bereits durch die Software Factory getroffen sind, und diese auch entsprechende Komponenten zur Implementierung dieser Architektur bereithält. Architekten und Entwickler können sich damit im Normalfall auf die Umsetzung der jeweiligen Fachlichkeit der Anwendung konzentrieren. Da die Anwendung jedoch nach wie vor manuell entwickelt (und nicht generiert wird), ist es aber jederzeit möglich, dort von den Vorgaben der Software Factory abzuweichen, wo dies erforderlich ist.

### 2.2. IsyFact: Standards und Erweiterungen

Die **IsyFact** ist eine freie allgemeine Software Factory zum Bau von IT-Systemen, die vom Bundesverwaltungsamt (BVA) bereitgestellt wird. Sie umfasst Standards und Erweiterungen.

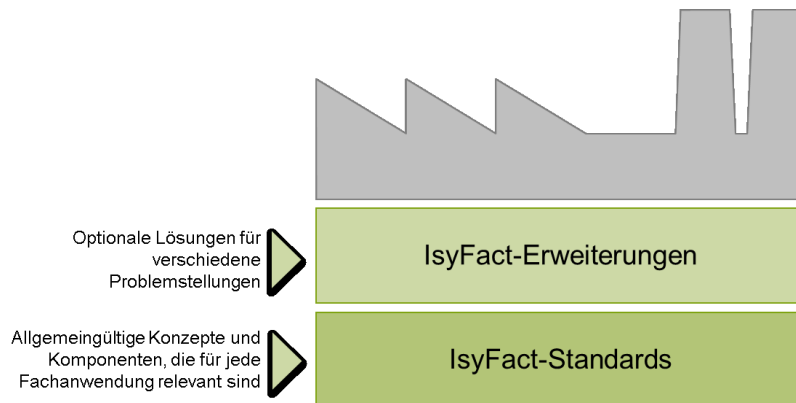


Abbildung 1: Die verschiedenen Ebenen der IsyFact

Die **IsyFact-Standards** bilden das architektonische, technologische und methodische Fundament der IsyFact. Sie umfassen allgemeingültige und wiederverwendbare Konzepte und Komponenten, die für die Entwicklung beliebiger Fachanwendungen relevant sind. Beispielsweise: Allgemeine Softwarereferenzarchitektur, Vorgaben zur Fehlerbehandlung, Logging etc.

Die **IsyFact-Erweiterungen** sind optionale wiederverwendbare Lösungen für verschiedene Problemstellungen, die aufgrund spezifischer Anforderungen des BVA oder seiner Kunden entwickelt wurden und auf den IsyFact-Standards aufbauen. Bspw.: Protokollierung, Regelwerk etc.

Die Artefakte der IsyFact-Standards und -Erweiterungen werden in gleicherweise in 5 Säulen klassifiziert, die der Strukturierung der Inhalte der Factory dienen.

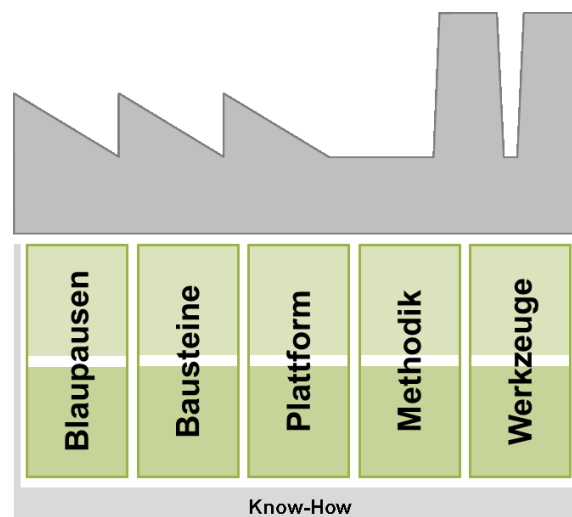


Abbildung 2: Die 5 Säulen der IsyFact

Die Einzelnen Säulen sind:

- **Blaupausen:** Die Blaupausen beschreiben die Architektur und Konzepte der Anwendungslandschaft.
- **Bausteine:** Die Bausteine der IsyFact sind wieder verwendbare Softwarelösungen.



- **Plattform:** Eine einheitliche Plattform über alle IT-Systeme, die einen standardisierten und effizienten Systembetrieb ermöglicht.
- **Methodik:** Grundlage für die Umsetzung von Systemen mit der IsyFact ist eine standardisierte Vorgehensweise nach dem V-Modell XT.
- **Werkzeuge:** Die IsyFact setzt auf Automatisierung und Werkzeugunterstützung bei der Erstellung von IT-Systemen. Dazu bietet sie vorkonfigurierte Werkzeuge für Modellierung, Programmierung, Installation, Tests oder die Fehlerverfolgung.

### 2.3. Nutzungsszenarien und Tailoring

Hinter der IsyFact steht der Gedanke, die Anwendungsentwicklung für eine bestimmten Domäne zu vereinfachen, indem man das Wissen über diese Domäne und die dort benötigten Anwendungen systematisiert und dokumentiert und in Form einer für die Domäne spezifischen Software Factory bereitstellt. Da spezifisches Domänenwissen jedoch per Definition nicht Teil der IsyFact sein kann, muss es aus dem jeweiligen Anwendungskontext heraus ergänzt werden. Dabei wird zunächst beschrieben, welche Teile der IsyFact für die jeweilige Domäne überhaupt relevant sind und zum Einsatz kommen sollen. Dann werden die Konzepte der IsyFact konkretisiert, und domänenspezifische Bausteine und Architekturen werden hinzugefügt. Diesen Vorgang bezeichnet man insgesamt als *Tailoring*. Das Resultat des Tailorings ist eine spezifische Factory für einen Anwendungskontext, wie sie z.B. beim Bundesverwaltungsamt in Form der Register Factory<sup>1</sup> für den Kontext der Register Anwendungen erstellt wurde.

Das Tailoring ist ein sinnvoller Schritt, um möglichst viel Wissen über die Entwicklung von Anwendungen in Form einer Factory vorzuhalten. Es ist für die Nutzung der IsyFact jedoch nicht erforderlich. Insgesamt gibt es vier mögliche Nutzungsszenarien, die in Abbildung 3 beschrieben und im Folgenden erläutert werden:

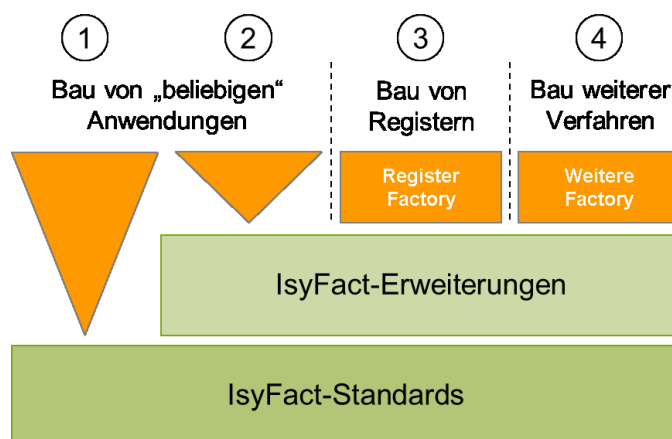


Abbildung 3: Nutzungsszenarien der IsyFact

<sup>1</sup> [www.register-factory.de](http://www.register-factory.de)

- **Szenario 1 – Direkte Nutzung der IsyFact-Standards:** Die direkte Nutzung des Standards ermöglicht es, von den Vorteilen der bewährten Standards der Factory unmittelbar zu profitieren. Dieses Szenario ist insbesondere immer dann sinnvoll, wenn einzelne Systeme und keine komplette Anwendungslandschaft gleichartiger zusammenhängender Systeme umgesetzt werden sollen.
- **Szenario 2 – Nutzung von IsyFact-Erweiterungen:** Die IsyFact-Erweiterungen sind fertige Speziallösungen, die in „beliebigen“ Verfahren zum Einsatz kommen können. Dies ist oft sogar dann möglich, wenn die IsyFact-Standards im jeweiligen Verfahren nicht angewandt werden. Der Einsatz der Standards wird jedoch empfohlen. In diesem Szenario findet ein „leichtgewichtiges Tailoring“ statt, in dem die relevanten Erweiterungen ausgewählt werden.
- **Szenario 3 – Nutzung einer bestehenden spezifischen Factory:** Wenn für denselben oder einen ähnlichen Anwendungskontext bereits eine spezifische Factory zur Verfügung steht (wie bspw. die Register Factory) kann diese direkt verwendet werden. Eine spezifische Factory ist eine durch Tailoring zugeschnittene Variante der IsyFact, die diese um spezifische Aspekte des jeweiligen Kontextes erweitert.
- **Szenario 4 – Definition einer neuen spezifischen Factory:** Die IsyFact ermöglicht und fördert die Definition von neuen spezifischen Factories. Dies ist dann sinnvoll, wenn eine große homogene Anwendungslandschaft in einem spezifischen Kontext aufgebaut werden soll, für den bisher keine spezifische Factory existiert.

Dabei empfiehlt es sich, nach einer initiale Analyse und Auswahl der einzusetzenden IsyFact-Komponenten im Weiteren iterativ vorzugehen: Parallel zur Entwicklung der ersten Fachanwendung auf der Basis der IsyFact fließen die dabei gewonnenen Erfahrungen in das Tailoring ein. So entstehen für nachfolgende Anwendungs-Entwicklungen eine spezifische Referenzarchitektur und die Komponenten, mit denen sie implementiert werden kann. Die folgenden Entwicklungen sind wiederum der Prüfstein für die Qualität und Reife der spezifischen Factory und liefern neue Erkenntnisse für ihre Fortschreibung. Auf diese Weise stabilisiert sich mit der Zeit die Factory für die eigene Anwendungsdomäne.

## 2.4. Festlegungen der IsyFact-Konformität

Durch die unterschiedlichen Nutzungsszenarien der IsyFact existieren mehrere Varianten, die den Grad der Verwendung von IsyFact in einer Anwendung beschreiben. Zur einfacheren Handhabung der Abhängigkeiten definieren wir folgende Begriffe, die in den Dokumenten der IsyFact verwendet werden:

- **IsyFact-konform** sind Anwendungen, die vollständig und ohne Ausnahmen auf den IsyFact-Standards aufbauen.

- **IsyFact-kompatibel** sind Anwendungen, deren Außenverhalten sich nach IsyFact-Standards richtet und ohne Anpassungen in einer Systemlandschaft mit IsyFact-konformen Anwendungen betreibbar ist.
- **IsyFact-basierend** sind Anwendungen, die Teile der IsyFact-Standards verwenden, aber nicht kompatibel sind.
- Eine **IsyFact-Anwendung** ist mindestens IsyFact-kompatibel und kann IsyFact-Erweiterungen nutzen. Im Prinzip ist jede IsyFact-Anwendung auch eine IsyFact-Erweiterung, bietet aber i.d.R. keine querschnittliche Funktionalität für andere Anwendungen an und ist dementsprechend nicht synonym zu verwenden.

## 2.5. Mitarbeit an der IsyFact

Das Bundesverwaltungsamt hat sich entschlossen, die IsyFact-Standards als Open Source zu veröffentlichen damit zum einen ein möglichst großer Kreis von Anwendern von der bisherigen Entwicklung profitiert. Langfristig ist das Ziel, zusätzlich einen Marktplatz zu etablieren, auf dem jeder Anwender seine Weiterentwicklungen und neuen Komponenten mit anderen Anwendern teilen kann.

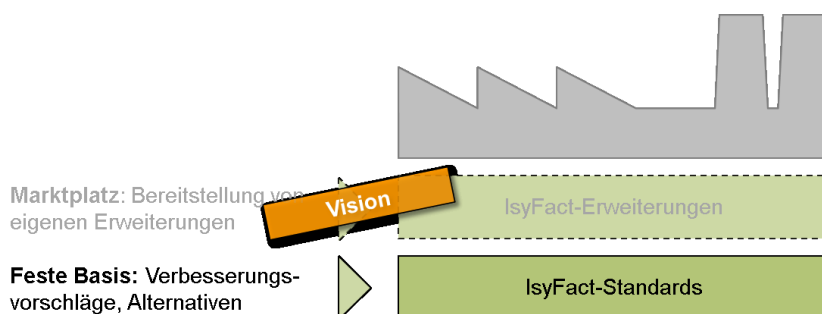


Abbildung 4: Mitarbeit an der IsyFact

Für die Beteiligung an der Weiterentwicklung der IsyFact gibt es zwei Möglichkeiten (siehe Abbildung 4):

- Die **IsyFact-Standards** bilden eine feste Basis, die nur vom BVA selbst weiterentwickelt wird. Anwender können aber jederzeit Verbesserungsvorschläge oder Alternativen für bestehende Komponenten einbringen. Das BVA wird diese Vorschläge sammeln, bewerten und konsolidiert in die Weiterentwicklung der IsyFact einfließen lassen. Diese moderierende Rolle ist wichtig, damit die gemeinsame Basis, auf der letztlich auch das Funktionieren der Factory und aller Erweiterungen beruht, erhalten bleibt.
- **Neue IsyFact-Erweiterungen** kann jeder Anwender beitragen, veröffentlichen und auf dem zukünftigen Marktplatz bereitstellen. Voraussetzung ist dabei lediglich, dass diese IsyFact-kompatibel sind. (Vision)

## 2.6. Historie der IsyFact

In diesem Abschnitt wird ein Einblick in die Entstehung der IsyFact gegeben. Dies ist relevant, da die vorhandenen Konzepte durch deren Historie geprägt wurden und sich darin teilweise noch „historisch bedingte“ Formulierungen befinden (mehr dazu in Abschnitt 3.1).

Die IsyFact ist aus der Register Factory entstanden, die eine Software Factory für die Entwicklung großer, Anwendungs- und Registerlandschaften im Behördenumfeld ist. Die Register Factory ist über mehrere Jahre hinweg beim Bundesverwaltungsamt (BVA) entwickelt worden und Grundlage für geschäftskritische Anwendungen verschiedener Behörden, deren Anwendungslandschaften das BVA nach den Vorgaben der Register Factory entwickelt hat und betreibt. Darüber hinaus stellt das BVA die Register Factory auf Anfrage auch anderen Behörden zu Verfügung, die darauf eigenständig ihre Anwendungen entwickeln und betreiben.

In diesem Kontext wird zwischen den Systemtypen *Register* und *Geschäftsanwendungen* unterschieden. Register dienen der Sammlung von Informationen und besitzen keine Präsentationslogik oder Prozesse. Geschäftsanwendungen implementieren fachliche Logik, Prozesse und präsentieren die Daten aus den Registern. Die Anwendungsarchitektur der Register Factory und deren Bausteine und Methoden ließen sich aber ohne weiteres zur Entwicklung beliebiger fachlicher Anwendungen (*Fachanwendungen*) benutzen. Geschäftsanwendungen und Register sind lediglich Spezialfälle solcher Fachanwendungen. Um die Artefakte der Register Factory auch in anderen Kontexten zur Verfügung zu stellen, wurden die bestehenden Artefakte der Register Factory umstrukturiert und neu aufgeteilt in IsyFact (Standards und Erweiterungen) und Register Factory.

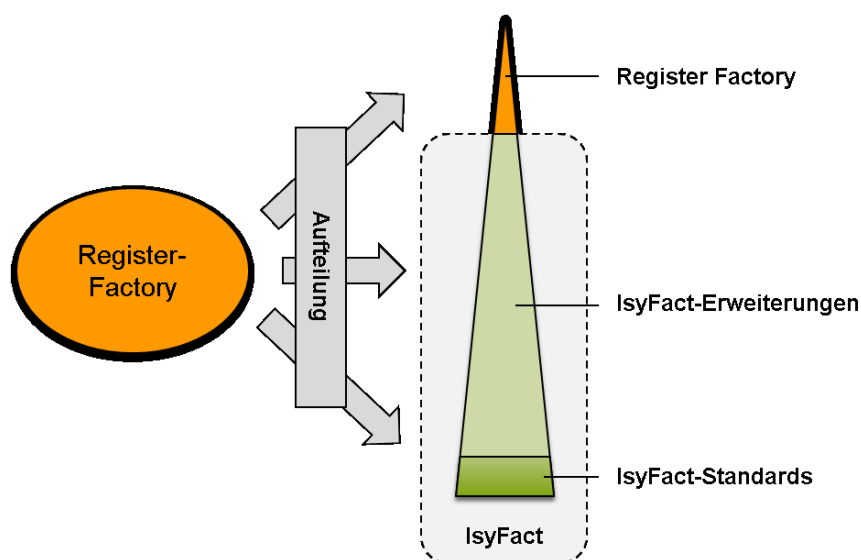


Abbildung 5: Aufteilung der Register Factory zur IsyFact

Die IsyFact enthält ausschließlich allgemeingültige „register-neutrale“ Konzepte und Komponenten. Die Register Factory ist seit deren

Umstellung eine spezifische Factory, die auf der IsyFact aufbaut (vergleiche die verschiedenen Nutzungsszenarien in Abschnitt 2.3). Sie wird durch das BVA ebenfalls aktiv weiterentwickelt.

### 3. Rahmenbedingungen

Die folgenden Abschnitte geben einen Überblick über die Rahmenbedingungen, unter denen die IsyFact entstanden ist und weiter entwickelt wird.

Die IsyFact wurde ursprünglich zur internen Verwendung beim Bundesverwaltungsamt entwickelt. Ihre Umstellung zu einer allgemeinen, von diesem Entstehungskontext losgelösten Software Factory ist ein Prozess, der noch nicht vollständig abgeschlossen ist.

#### 3.1. Auswirkung der Historie der IsyFact

Auf Grund der Historie der IsyFact (siehe Abschnitt 2.6), besitzen die Artefakte der IsyFact teilweise noch Bezug zum Kontext der Register Factory: Die enthaltenen Dokumente der IsyFact beschreiben allgemein die Entwicklung von Fachanwendungen. Die verwendeten Beispiele stammen jedoch zum Teil aus dem Kontext der Registeranwendungen. Daher finden sich Begriffe wie „Register“ vielfach noch in Beispielquelltexten wieder. Ähnliches gilt auch für Pfadangaben, Parameternamen oder Variablen, die insbesondere in den Konzepten für den Betrieb der Systemlandschaft auftauchen. Die entsprechenden Bezeichner sind nicht als zwingende Vorgaben zu verstehen, sondern spiegeln einfach die Historie und den aktuellen Stand der IsyFact wieder. Unter anderem wurden bestehende Konventionen auch deshalb nicht verändert, um die Konsistenz der Dokumentation mit bestehenden IsyFact-konformen Systemlandschaften zu wahren.

Eine andere, ebenfalls historisch bedingte Bezeichnung für eine IsyFact-Systemlandschaft ist „Plattform für Informationssysteme“, kurz *PLIS*. Diese Abkürzung findet sich als Präfix in den Namen der IsyFact-Java-Bibliotheken wieder, z.B. in *plis-web*, *plis-persistence* oder *isy-logging*.

#### 3.2. Aktueller Stand und Weiterentwicklung

Die veröffentlichten IsyFact-Standards bilden ein umfassendes Fundament für den effizienten Bau und Betrieb homogener Anwendungen. Darauf aufbauend sind als nächstes die folgenden Schritte geplant.

**Veröffentlichung weiterer Standards und Erweiterungen.** Die Veröffentlichung weiterer Standards und Erweiterungen ist geplant, erfordert jedoch eine Überarbeitung und Qualitätskontrolle, die nur schrittweise erfolgen kann. Aus diesem Grund werden zunächst die IsyFact-Standards veröffentlicht, später dann nach und nach Erweiterungen, sofern deren Veröffentlichung möglich ist und diese für andere Kontexte von Nutzen sind. Die Dokumentation der IsyFact-Standards referenziert an einigen Stellen auf Bausteine der IsyFact-Erweiterungen. Diese Referenzen wurden, im Vorgriff auf die bevorstehende Veröffentlichung der Erweiterungen, in der Dokumentation belassen.

Bisher unveröffentlichte Erweiterungen können Bundesbehörden im Rahmen von Verwaltungsvereinbarungen und anderen Behörden im Rahmen der Kieler Beschlüsse auf Anfrage bereitgestellt werden.

- **Anpassung der Terminologie.** Langfristig ist es geplant die in Abschnitt 3.1 angesprochenen Bezeichner anzupassen. Vorrang hat hierbei jedoch die Kompatibilität zu bestehenden Systemlandschaften, die mit der IsyFact bereits erstellt wurden.
- **Einführung eines Marktplatzes.** Die Einführung des in Abschnitt 2.5 beschriebenen Marktplatzes ist ebenfalls ein langfristiges Ziel.

### 3.3. Verwendete Software-Produkte

Die IsyFact basiert auf einer Reihe von etablierten Software-Produkten, die die unterschiedlichen funktionalen Anforderungen eines Anwendungssystems realisieren. In den meisten Fällen sind dies kostenfreie Open-Source-Lösungen, in einigen Fällen, z. B. im Bereich Datenbanken, wird jedoch auch auf **kommerzielle Produkte** verwiesen. In solchen Fällen beziehen sich auch ggf. mitgelieferte Anleitungen und Skripte auf diese kommerziellen Produkte. Der Einsatz des jeweils genannten Produktes ist zwar in IsyFact vorgesehen, aber der Einsatz alternativer Produkte sollte mit überschaubarem Aufwand möglich sein.

Wenn Sie uns eine Ergänzung zum jeweiligen Konzept zukommen lassen, die den Einsatz eines alternativen kostenpflichtigen oder kostenfreien Produkts beschreibt, werden wir die Aufnahme in den Standard prüfen.

Unser Ziel ist es, einen möglichst „freien“ Standard zu etablieren (sowohl kostenfrei als auch Open-Source), der zwar einheitliche Vorgaben definiert, aber auch Spielräume lässt, wo diese sinnvoll und möglich sind.

### 3.4. Annahmen zu Projektrollen

Die IsyFact ermöglicht den Betrieb der Systeme einer Anwendungslandschaft auf einer gemeinsamen Plattform. Die einzelnen Anwendungen werden dabei meist in getrennten Projekten entwickelt. Projekte können dabei sowohl sequentiell als auch parallel ablaufen. Die Factory garantiert dabei, dass die Anwendungen zum einen auf der Plattform betreibbar sind und dass sie zum anderen effizient und nach einheitlichen Standards entwickelt werden.

Durch die gemeinsame Plattform und die Schnittstellen der Anwendungen untereinander ergeben sich Abhängigkeiten zwischen den Projekten. Aus organisatorischer Sicht handelt es sich dabei um ein Multiprojekt, für das eine geeignete Struktur mit entsprechenden Rollen zu schaffen ist. Diese kann nicht im Rahmen der IsyFact vorgegeben werden, sondern muss in jedem Umfeld, in dem die IsyFact eingesetzt wird, nach den dort geltenden Regeln definiert werden. Allerdings macht die IsyFact an einigen Stellen Annahmen darüber, welche Rollen es im jeweiligen Projekt gibt und welche Verantwortlichkeiten diesen Rolle zugeordnet sind. Beispiele hierfür sind

die Verantwortung für die Einhaltung der Architektur bzw. die Entscheidungskompetenz, davon abzuweichen.

Im Folgenden werden die verschiedenen Rollen und deren Verantwortlichkeiten aufgeführt, die in den Konzepten verwendet werden. Die jeweiligen Aufgaben sind durch die entsprechende Rolle im konkreten Projektkontext zu übernehmen:

- **Chefarchitekt:** Der Chefarchitekt verantwortet den adäquaten Technikeinsatz und die Architektur im Gesamtprojekt bzw. auf Ebene der Anwendungslandschaft.
- **Fachlicher Architekt:** Der Fachliche Architekt verantwortet die Struktur der einzelnen Systeme und Querschnittskomponenten in einer Anwendungslandschaft aus fachlicher Sicht.
- **Systemarchitekt (Technischer Chefdesigner):** Die Systemarchitekten, oder auch Technische Chefdesigner genannt, verantworten den adäquaten Technikeinsatz und die Architektur in einem Teilprojekt bzw. für eines oder mehrere IT-Systeme.
- **SW-Entwickler:** Die SW-Entwickler sind zuständig für die Realisierung der IT-Systeme.
- **Change Control Board:** Das Change Control Board ist ein Gremium, das bei wichtigen Änderungen einberufen wird und entscheidet, wie über eine oder mehrere zusammenhängende Änderungen verfahren werden soll.
- **Architekturboard:** Das Architekturboard ist ein Gremium, welches die konzeptionelle Weiterentwicklung einer spezifischen Factory steuert. Es tritt regelmäßig zusammen, um aktuelle Anforderungen und Problemstellungen zu diskutieren und die langfristige Tragfähigkeit der Factory sicherzustellen.

### 3.5. Styleguides und die Gestaltung von Benutzeroberflächen

Bei der Entwicklung einer Anwendungslandschaft sollten nicht nur die Architektur der einzelnen Anwendungen sondern auch die Benutzeroberflächen einheitlichen Standards folgen. Die Standards für die Benutzeroberflächen werden üblicherweise durch einen Styleguide vorgegeben, der u.a. beschreibt, welche Elemente eine graphische Benutzeroberfläche besitzt, wie diese zu gestalten sind und wie sie miteinander kombinieren werden, um bestimmte Funktionen zu realisieren.

Die Dokumente der IsyFact verweisen an verschiedenen Stellen auf den Styleguide und dort zu definierende Regeln. Es wird jedoch vorausgesetzt, dass jeder Anwender der IsyFact einen eigenen Styleguide erstellt, der spezifisch auf seinen Anwendungskontext abgestimmt ist.



## 4. IsyFact-Standards

Im Folgenden sind die unterschiedlichen Vorgaben und Komponenten der IsyFact-Standards beschrieben. Der vorliegende Abschnitt ist als Referenz und als schneller Einstieg in die einzelnen Bestandteile von IsyFact konzipiert. Die Unterabschnitte enthalten jeweils eine kurze Erläuterung zu der jeweiligen Komponente und verweisen dann auf die zugehörige Dokumentation.

Der Aufbau dieses Kapitels orientiert sich am Entwicklungsprozess einer Fachanwendung und macht Vorgaben zu den folgenden Phasen des V-Modells XT: Spezifikation, Systementwurf und Realisierung. Darüber hinaus werden die querschnittlich zu nutzenden Basiskomponenten beschrieben.

### 4.1. Vorgaben für die Spezifikation

Für die Systemspezifikation ist festgelegt, welche Inhalte und welche Form diese haben soll. Die Vorgaben dazu befinden sich mitsamt Beispielen in der Dokumentvorlage für die Systemspezifikation [IsyFact-SystemspezifikationVorlage]. Bereits bei der Systemspezifikation ist auf die Einhaltung der [IsyFact-Namenskonvention] zu achten.

Für die Erfassung von Anforderungslisten ist ebenfalls eine Vorlage vorhanden [IsyFact-Anforderungsliste]. Die Anforderungsliste ist ein Instrument, um die Übersicht und die Nachvollziehbarkeit des Umsetzungsstatus aller Anforderungen an ein System im gesamten Projektlebenszyklus zu erhalten. Weitere Erläuterungen finden sich in der Vorlage.

Im Dokument [IsyFact-Datenflussdiagramme] wird die Erstellung von Datenflussdiagrammen beschrieben und eine Leseanleitung dazu geliefert.

#### Verweise

[IsyFact-SystemspezifikationVorlage]

Vorlage Systemspezifikation

[40\\_Methodik\10\\_Systemspezifikation\IsyFact-Vorlage\\_Systemspezifikation.dotx](#)

[IsyFact-Namenskonventionen]

IsyFact-Namenskonventionen

[40\\_Methodik\00\\_Allgemein\IsyFact-Namenskonventionen.pdf](#)

[IsyFact-Anforderungsliste]

Vorlage für Anforderungslisten

[40\\_Methodik\10\\_Systemspezifikation\IsyFact-Vorlage-Anforderungsliste.xlsx](#)

[IsyFact-Datenflussdiagramme]

Anleitung zur Erstellung und zum Lesen von Datenflussdiagrammen

[40\\_Methodik\10\\_Systemspezifikation\Anleitung\\_zu\\_Datenflussdiagrammen.docx](#)

## 4.2. Vorgaben für den Systementwurf

Der Systementwurf ist gemäß der Vorlage zum Systementwurf zu erstellen [IsyFact-SystementwurfVorlage]. Diese legt die äußere Form und die Gliederung des Dokuments fest. Darin ist auch das Vorgehen zur Modellierung und die zu beschreibenden Inhalte vorgegeben.

Zur Konstruktion des Systems, also den eigentlichen Inhalten des Systementwurfs, existieren eine Reihe von Dokumenten, die inhaltliche Vorgaben machen. Grundlage ist das Dokument zur Referenzarchitektur, das die wesentlichen Inhalte auf grober Ebene festlegt [IsyFact-Referenzarchitektur]. Dieses Dokument wird durch weitere Dokumente konkretisiert. Die wichtigsten Entscheidungen zu den zu nutzenden Produkten sind in [IsyFact-Produktkatalog] festgelegt. Anwendungen müssen auf Basis dieser Produkte und Bibliotheken gebaut werden.

Weitere Detaillierungen der Referenzarchitektur werden in den folgenden drei Abschnitten erläutert. Jeder dieser Abschnitte fokussiert dabei auf eine spezielle Sicht der zu erstellenden Architektur: Fachliche Architektur (A-Architektur), Technische Architektur (T-Architektur) und Technische Infrastruktur (TI-Architektur). Die genaue Unterscheidung zwischen diesen Sichten ist in [IsyFact-Referenzarchitektur] beschrieben.

### Verweise

- [IsyFact-SystementwurfVorlage]  
Vorlage Systementwurf  
[40 Methodik\20 Systementwurf\IsyFact-Vorlage Systementwurf.dotx](#)
- [IsyFact-Referenzarchitektur]  
IsyFact – Referenzarchitektur  
[00 Allgemein\IsyFact-Referenzarchitektur.pdf](#)
- [IsyFact-Produktkatalog]  
IsyFact Produktkatalog  
[00 Allgemein\Produktkatalog.xlsx](#)

### 4.2.1 Vorgaben für die A-Architektur

Das Dokument [IsyFact-Referenzarchitektur] gibt die Grundkonzepte für den Aufbau einer Anwendung vor. Dazu ist eine Referenzarchitektur vorgegeben. Diese allgemeine Referenzarchitektur muss für die zu bauenden Systeme in einer fachlichen Architektur konkretisiert werden.

Die Anwendungsarchitektur strukturiert die Software dabei in Anwendungen, die jeweils bestimmte fachliche Geschäftsprozesse unterstützen. Diese fachliche Architektur wird in der softwaretechnischen Architektur abgebildet. Der Fachliche Chefarchitekt definiert dabei, aus welchen Komponenten sich die Anwendungslandschaft zusammensetzt, wie diese miteinander oder mit Systemen außerhalb der Anwendungslandschaft über Schnittstellen agieren und welchen fachlichen Domänen diese Systeme zugehörig sind.

Die Facharchitektur beschreibt zudem die Geschäftsprozesse, Funktionalitäten und Regeln, nach denen die unterstützenden Systeme strukturiert sind. Daher hält sie eine aktuelle und vollständige Dokumentation dieser vor.

## Verweise

[IsyFact-Referenzarchitektur]

IsyFact – Referenzarchitektur

[00\\_Allgemein\IsyFact-Referenzarchitektur.pdf](#)

### 4.2.2 Vorgaben für die T-Architektur

Im Dokument [IsyFact-Referenzarchitektur-IT-System] wird eine Technische Referenzarchitektur für IT-Systeme eingeführt und erläutert. Danach besteht ein IT-System aus den Komponenten, die in Abbildung 6 dargestellt sind.

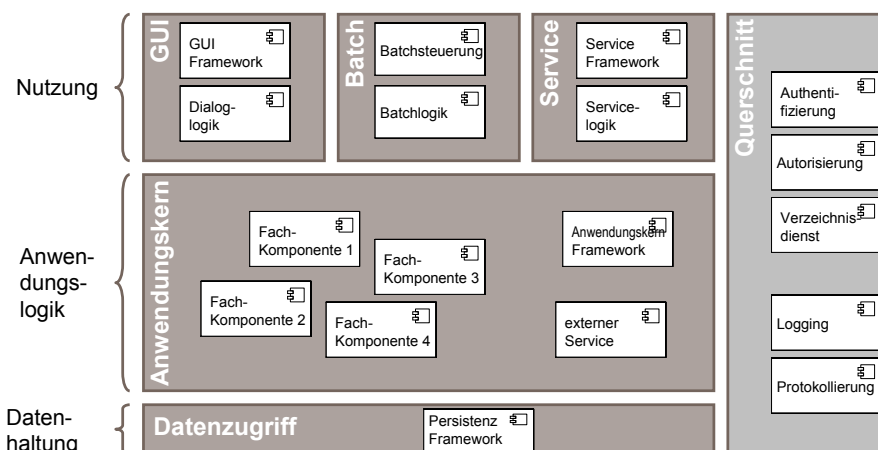


Abbildung 6: Technische Referenzarchitektur

Zu den einzelnen Komponenten dieser Architektur gibt es detaillierte Konzepte, Basisimplementierungen und Codebeispiele [Vorlage-Anwendung], wie diese umzusetzen sind.

## Verweise

[IsyFact-Referenzarchitektur-IT-System]

IsyFact – Referenzarchitektur IT-System

[00\\_Allgemein\IsyFact-Referenzarchitektur-IT-System.pdf](#)

[Vorlage-Anwendung]

Beispielimplementierung „Vorlage-Anwendung“  
wird auf Anfrage bereitgestellt

Die Komponenten und die erläuternden Konzepte sind die folgenden:

#### 4.2.2.1 Datenzugriff

Der Datenzugriff erfolgt über JPA bzw. Hibernate. Die genaue Verwendung von JPA/Hibernate und die Prinzipien, nach denen bei der Persistierung von Datenobjekten vorgegangen werden soll, sind im Dokument [DatenzugriffDetailkonzept] festgelegt.

## Verweise

[DatenzugriffDetailkonzept]

Detailkonzept Komponente Datenzugriff

[10\\_Blaupausen\technische\\_Architektur\Detailkonzept\\_Komponente\\_Datenzugriff.pdf](#)

### 4.2.2.2 Anwendungskern

Der Anwendungskern ist in Komponenten aufgebaut. Diese Komponenten werden durch Spring konfiguriert. Der Schnitt der Komponenten ist durch die fachliche Referenzarchitektur vorgegeben. Die Ergänzung durch technische Komponenten und die Vorgaben für die Nutzung von Spring zu deren Konfiguration sind in [AnwendungskernDetailkonzept] festgelegt.

## Verweise

[AnwendungskernDetailkonzept]

Detailkonzept Komponente Anwendungskern

[10\\_Blaupausen\technische\\_Architektur\Detailkonzept\\_Komponente\\_Anwendungskern.pdf](#)

### 4.2.2.3 Batch

Vorgaben für Batches befinden sich im Dokument [BatchDetailkonzept]. Hier ist auch der zu verwendende Batchrahmen beschrieben.

## Verweise

[BatchDetailkonzept]

Detailkonzept Komponente Batch

[10\\_Blaupausen\technische\\_Architektur\Detailkonzept\\_Batch.pdf](#)

### 4.2.2.4 Service

Die Art und Weise, wie eine Anwendung einen Service zur Nutzung innerhalb einer Plattform bereitstellt, ist in [ServiceDetailkonzept] dargestellt.

Innerhalb einer Plattform kommunizieren die einzelnen IT-Systeme mittels der Technologie Spring HttpInvoker, die eine besonders effiziente Kommunikation über http umsetzt. Die Grundlagen dazu sind in [ServicekommunikationKonzept] beschrieben.

## Verweise

[ServicekommunikationKonzept]

Grundlagen der Servicekommunikation innerhalb der Plattform

[10\\_Blaupausen\Integrationsplattform\Grundlagen\\_interne\\_Servicekommunikation.pdf](#)

[ServiceDetailkonzept]

Detailkonzept Komponente Service

[10\\_Blaupausen\technische\\_Architektur\Detailkonzept\\_Komponente\\_Service.pdf](#)

#### 4.2.2.5 GUI

Die Entwicklung von GUIs erfolgt als Web-GUI auf Basis von JSF und Spring WebFlow. Die genauen Festlegungen zur Web-GUI-Entwicklung finden sich im Dokument [GUIDetailkonzept].

#### Verweise

[GUIDetailkonzept]

Detailkonzept Komponente GUI

[10\\_Blaupausen\technische\\_Architektur\Detailkonzept\\_Komponente\\_Web\\_GUI.pdf](#)

#### 4.2.3 Vorgaben für die TI-Architektur

Die TI-Architektur von Fachanwendungen ist in [IsyFact-Referenzarchitektur] beschrieben. Die zu verwendende Infrastruktur ist z. T. durch den [IsyFact-Produktkatalog] vorgegeben.

#### Verweise

[IsyFact-Referenzarchitektur]

IsyFact – Referenzarchitektur

[00\\_Allgemein\IsyFact-Referenzarchitektur.pdf](#)

[IsyFact-Produktkatalog]

IsyFact – Produktkatalog

[00\\_Allgemein\Produktkatalog.xlsx](#)

#### 4.3. Vorgaben für die Realisierung

Die Realisierung hat gemäß den Programmierkonventionen zu erfolgen [JavaProgrammKonv].

Die Bibliotheken der IsyFact werden als JAR-Dateien bereitgestellt.

Darüber hinaus existiert eine Vorlage-Anwendung [Vorlage-Anwendung], die jedoch in ihrer aktuellen Form kein Teil der IsyFact-Standards ist und nur auf Anfrage bereitgestellt wird. Hintergrund ist, dass die Vorlage-Anwendung veraltet ist und es sich noch um eine Register-Anwendung handelt. Darin enthalten sind:

- CD-Register (Vorlage-Register)
- CD-Geschäftsanwendung mit Web-GUI (Vorlage-Geschäftsanwendung)
- Service Gateway für ein- und ausgehende Kommunikation (Vorlage-Register-Service-Consumer) bzw. (Vorlage-Register-Service-Provider)

Eine aktualisierte Vorlage-Anwendung auf Basis der IsyFact ist derzeit in der Entwicklung.

Neben der Beispielanwendung gibt es mit dem IsyFact-Tutorial [IsyFact-Tutorial] eine weitere Handreichung, um sich in die Implementierungsvorgaben einzuarbeiten.

Für jedes realisierte System ist ein Handbuch für den Systembetrieb zu erstellen. Eine Vorlage dafür ist in [IsyFact-Systemhandbuch] enthalten.

## Verweise

- [JavaProgrammKonv]
  - Java-Programmierkonventionen
  - [40 Methodik\30 Implementierung\Java-Programmierkonventionen.pdf](#)
- [Vorlage-Anwendung]
  - Vorlage-Anwendung
  - wird auf Anfrage bereitgestellt*
- [IsyFact-Tutorial]
  - IsyFact – Tutorial
  - [00 Allgemein\IsyFact-Tutorial.pdf](#)
- [IsyFact-Systemhandbuch]
  - Vorlage Systemhandbuch
  - [40 Methodik\40 Betrieb\30 Implementierung\IsyFact-Vorlage\\_Systemhandbuch.dot](#)

## 4.4. Querschnittlich zu nutzende Basiskomponenten

Die IsyFact-Standards stellen eine Reihe von Basiskomponenten zur Umsetzung querschnittlicher Funktionalitäten bereit, die für alle IT-Systeme relevant und zu nutzen sind. Diese werden im Folgenden dargestellt:

### 4.4.1 Fehlerbehandlung

Im Dokument [FehlerbehandlungKonzept] ist beschrieben, in welchen Fällen und in welcher Form die Fehler- und Ausnahmebehandlung stattfinden soll.

## Verweise

- [FehlerbehandlungKonzept]
  - Konzept Fehlerbehandlung
  - [20\\_Bausteine\Fehlerbehandlung\Konzept\\_Fehlerbehandlung.pdf](#)

### 4.4.2 Administrative Überwachung und Konfiguration

Das Konzept [ÜberwachungKonfigKonzept] beschreibt, welche Arten von Konfiguration für eine Anwendung vorgesehen sind und wie diese umgesetzt werden sollen. Weiterhin wird in diesem Dokument gezeigt, wie die Überwachung und Administration einer Anwendung seitens des Systembetriebs erfolgt und welche Schnittstellen dazu durch die Anwendung zur Verfügung gestellt werden müssen.

## Verweise

[ÜberwachungKonfigKonzept]  
Konzept Überwachung und Konfiguration  
[20 Bausteine\Ueberwachung Konfiguration\Konzept Ueberwachun  
g-Konfiguration.pdf](#)

### 4.4.3 Behandlung von internationalen Sonderzeichen

Fachanwendungen müssen zum Teil mit Einträgen umgehen, die nicht den geläufigen Zeichenstandards und Codierungen unterliegen. Im Dokument [SonderzeichenKonzept] werden Festlegungen getroffen, wie mit daraus resultierenden Problemstellungen umgegangen wird. In diesem Zusammenhang müssen oft auch Namen transkribiert werden. Die dafür zu verwendenden Regeln sind ebenfalls im Dokument enthalten.

## Verweise

[SonderzeichenKonzept]  
Konzept Umgang mit Sonderzeichen  
[20 Bausteine\Sonderzeichen\Konzept Umgang mit Sonderzeichen  
.pdf](#)

### 4.4.4 Logging

Das Konzept [LoggingKonzept] beschreibt, mit welchen technischen Mitteln, in welchen Formaten und zu welchen Zwecken das Logging in einer Anwendung erfolgen soll.

## Verweise

[LoggingKonzept]  
Konzept Logging  
[20 Bausteine\Logging\Konzept Logging.pdf](#)

### 4.4.5 Berechtigungen

Zum Zugriff auf Informationen zu Berechtigungen eines Nutzers ist die Komponente „Sicherheit“ zu nutzen. Die Benutzung ist unter [SicherheitNutzerdok] beschrieben. Zur Nutzung dieser Komponente ist es erforderlich, dass die Rollen und Rechte einer Anwendung in einem speziellen Format abgelegt werden. Ein XML-Schema dazu findet sich im Anhang von [SicherheitNutzerdok].

## Verweise

[SicherheitNutzerdok]  
Nutzerdokumentation Komponente Sicherheit  
[20 Bausteine\Sicherheitskomponente\Nutzerdokumentation Sicherh  
eit.pdf](#)

### 4.4.6 LDAP-Zugriffe

Der Zugriff auf Daten in einem LDAP-Verzeichnis ist im Konzept [SpringLDAPNutzungskonzept] beschrieben.

## Verweise

[SpringLDAPNutzungskonzept]

Nutzungskonzept Spring-LDAP

[20\\_Bausteine\LDAP-Zugriffe\Nutzungskonzept\\_Spring\\_LDAP.pdf](#)

### 4.4.7 Polling

In Fachanwendungen müssen manchmal Polling-basierte Schnittstellen angesprochen werden. Polling bedeutet, dass in regelmäßigen Intervallen neue Daten zur Verarbeitung abgeholt werden sollen. Die Schnittstellen nutzen unterschiedliche technische Verfahren wie IMAP, Web-Services, HTTP-Invoker oder proprietäre Datenbank-basierte Schnittstellen, weitere sind denkbar. Aus Gründen der Ausfallsicherheit soll die Abholung der Daten von mehreren Instanzen einer Anwendung durchgeführt werden. Diese Instanzen müssen synchronisiert werden, so dass Nachrichten nicht mehrfach verarbeitet werden. Die zugrunde liegenden Schnittstellen-Technologien bieten dafür kein Standardverfahren an. Der Baustein Polling definiert ein solches Verfahren. Die Nutzerdokumentation befindet sich unter [PollingNutzerDok].

## Verweise

[PollingNutzerDok]

Polling Nutzerdokumentation

[20\\_Bausteine\Polling\Nutzerdokumentation\\_Polling.pdf](#)

### 4.4.8 Util

Die Bibliothek isy-util bietet nützliche Hilfsmittel, die von den Anwendungen der IsyFact genutzt werden können. Es handelt sich dabei um kleinere Utility-Klassen, welche die Implementierung vereinfachen. Diese werden in [NutzungskonzeptUtil] überblicksartig beschrieben.

## Verweise

[NutzungskonzeptUtil]

Nutzungskonzept isy-util

[20\\_Bausteine\Util\Nutzungskonzept isy-util.pdf](#)



## 5. IsyFact-Erweiterungen

Aktuell werden keine Erweiterungen unter einer Open-Source-Lizenz zur Verfügung gestellt (siehe auch Abschnitt 3.2). Auf den Webseiten der Register Factory befindet sich eine Übersicht über vorhandene Bausteine<sup>2</sup>, die in Zukunft potentiell als IsyFact-Erweiterung zur Verfügung gestellt werden. Teilweise können diese auf Anfrage herausgegeben werden. Bei Interesse, wenden Sie sich bitte per E-Mail an uns.

---

2

[http://www.bva.bund.de/DE/Organisation/Abteilungen/Abteilung\\_BIT/Leistungen/IT\\_Produkte/RegisterFactory/Produkt/dossier-produkte.html](http://www.bva.bund.de/DE/Organisation/Abteilungen/Abteilung_BIT/Leistungen/IT_Produkte/RegisterFactory/Produkt/dossier-produkte.html)

## 6. Erste Schritte

IsyFact ist modular aufgebaut und gibt keine strikte Lese-Reihenfolge vor. Um die IsyFact als Ganzes kennenzulernen, zu verstehen und einzusetzen, empfehlen wir das Folgende an den 5 Säulen orientierte Vorgehen:

1. **IsyFact-Einstieg:** Das Einstiegsdokument sollte im Ganzen gelesen werden, um einen Einblick in die Grundideen der Factory und die existierenden Rahmenbedingungen zu erhalten.
2. **IsyFact-Tutorial:** Das Tutorial gibt einen Überblick über wichtige Aspekte der IsyFact-konformen Anwendungsentwicklung. Es erleichtert dem Nutzer, die für ihn relevanten Aspekte der IsyFact zu identifizieren, die im Detail durchgearbeitet werden sollten.
3. **Blaupausen:** Die Blaupausen beschreiben auf mehreren Ebenen die Architektur, die allen anderen Konzepten der IsyFact zugrunde liegt und führen häufig verwendete Begriffe ein. Es bietet sich daher an, bei der Einarbeitung mit dieser „Säule“ zu beginnen.
4. **Bausteine:** Danach sollten die *relevanten* Bausteine im Detail gelesen werden.
5. **Plattform und Methodik:** Plattform und Methodik sollten wiederum im Ganzen gelesen werden, da sie querschnittlich die Basis des Systembetriebs bzw. der angewandten Methodik beschreiben.
6. **Werkzeuge:** Die Werkzeuge sind erst bei der tatsächlichen Realisierung der Systeme relevant. Es ist jedoch sinnvoll sich im Voraus einen Überblick zu verschaffen.
7. **Tailoring:** Das Tailoring-Dokument bietet einen Überblick über die Zuschneidbarkeit der Factory. Dies ist insbesondere dann relevant, wenn eine eigene Factory etabliert werden soll.

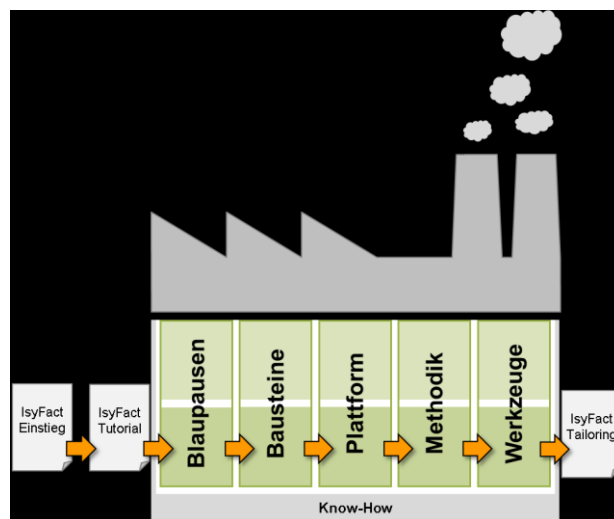


Abbildung 7: Einarbeitung in die IsyFact

## 7. Abbildungsverzeichnis

Abbildung 1: Die verschiedenen Ebenen der IsyFact.....	8
Abbildung 2: Die 5 Säulen der IsyFact .....	8
Abbildung 3: Nutzungsszenarien der IsyFact.....	9
Abbildung 4: Mitarbeit an der IsyFact.....	11
Abbildung 5: Aufteilung der Register Factory zur IsyFact.....	12
Abbildung 6: Technische Referenzarchitektur.....	19
Abbildung 7: Einarbeitung in die IsyFact .....	26