



Bundesverwaltungsamt



## IsyFact-Standard

# IsyFact – Referenzarchitektur

**Version 1.12**  
**01.03.2017**



„ des Bundesverwaltungsamts ist lizenziert unter einer Creative Commons Namensnennung 4.0 International Lizenz.



„IsyFact – Referenzarchitektur“  
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-Referenzarchitektur.docx
----------------	----------------------------------

# Inhaltsverzeichnis

<b>1. Einleitung.....</b>	<b>6</b>
1.1. Kurzüberblick .....	6
1.1.1 Fachliche Referenzarchitektur .....	7
1.1.2 Software Referenzarchitektur .....	8
1.1.3 Referenzarchitektur der technischen Infrastruktur .....	9
1.2. Inhaltsübersicht.....	11
<b>2. Übersicht.....</b>	<b>12</b>
2.1. Die IsyFact .....	12
2.1.1 Die Idee der IsyFact.....	12
2.1.2 Durch die IsyFact abgedeckte Bereiche .....	13
2.2. Einordnung der Referenzarchitektur in die IsyFact .....	14
2.3. Die Idee der Referenzarchitektur .....	14
2.4. Die Ziele der Referenzarchitektur .....	15
2.5. Tailoring der Referenzarchitektur .....	15
<b>3. Die fachliche Referenzarchitektur .....</b>	<b>17</b>
3.1. Die Strukturierung der Anwendungslandschaft .....	17
3.2. Fachliche Referenzarchitekturen für Anwendungssysteme .....	18
<b>4. Die Software-Referenzarchitektur.....</b>	<b>20</b>
4.1. Strukturierung der Register-Plattform .....	20
4.2. Nutzungsarten eines Anwendungssystems.....	22
4.3. Verwendung von Produkten.....	24
<b>5. Die Referenzarchitektur der technischen Infrastruktur .....</b>	<b>26</b>
5.1. Umgebungen .....	26
5.1.1 Produktionsumgebung (PRU).....	27
5.1.2 Staging-Umgebung (STU) .....	29
5.1.3 Externe Schulungsumgebung (XSU).....	31
5.1.4 Externe Testumgebung (XTU).....	31
5.1.5 Abnahmetestumgebung (ATU).....	32

5.1.6	Entwicklungstestumgebung (ETU) .....	32
5.2.	Minimalanforderungen an die Ablaufumgebung .....	33
<b>6.</b>	<b>Betriebliche Aspekte.....</b>	<b>34</b>
6.1.	Verfügbarkeit und Ausfallsicherheit .....	34
6.2.	Performance .....	35
6.3.	Lastverteilung.....	35
6.4.	Auslieferung, Installation und Versionswechsel .....	36
6.5.	Betriebsüberwachung .....	37
6.6.	Daten und Datensicherungen .....	38
6.6.1	Software und Konfigurationsdateien.....	38
6.6.2	Log-Dateien .....	38
6.6.3	Datenbank .....	38
<b>7.</b>	<b>Vereinfachte Varianten der Referenzarchitektur .....</b>	<b>40</b>
7.1.	Einführung.....	40
7.2.	Vereinfachung durch Verzicht auf eine SOA-Plattform .....	40
7.3.	Vereinfachungen in der technischen Infrastruktur .....	41
7.3.1	Installation ohne Clusterung.....	42
7.3.2	Verzicht auf das SAGA-Zonenmodell.....	42
7.4.	Erweiterungsmöglichkeiten einer vereinfachten Architektur.....	43
7.4.1	Installation innerhalb einer SAGA-konformen Plattform.....	43
7.4.2	Installation innerhalb eines Clusters.....	43
7.4.3	Installation innerhalb einer SOA .....	43
7.5.	Anwendungsbeispiel: Kleinverfahren.....	43
<b>8.</b>	<b>Quellenverzeichnis .....</b>	<b>45</b>
<b>9.</b>	<b>Abbildungsverzeichnis .....</b>	<b>47</b>

## 1. Einleitung

Die **IsyFact** ist eine freie allgemeine Software Factory zum Bau von IT-Systemen, die vom Bundesverwaltungsamt bereitgestellt wird. Sie wurde entwickelt, um unterschiedliche Fachanwendungen effizient entwerfen, realisieren und betreiben zu können. Der Begriff Factory steht für die schnelle Erzeugung neuer Anwendungen durch Anpassung und Ausgestaltung vorgefertigter Elemente, die in eine standardisierte Plattform integriert werden können.

Die IsyFact besteht aus den IsyFact-Standards und IsyFact-Erweiterungen:

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. Bspw.: 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 in diesem Dokument beschriebene Referenzarchitektur ist ein zentraler Teil der IsyFact-Standards und erläutert auf einer allgemeinen Ebene, wie eine Anwendungslandschaft zu beschreiben und umzusetzen ist.

### 1.1. Kurzüberblick

Dieser Abschnitt gibt einen zusammenfassenden Überblick über die wesentlichen Inhalte der Referenzarchitektur, die in späteren Kapiteln noch einmal ausführlich dargestellt werden. Dabei wird die Architektur aus drei verschiedenen Sichten betrachtet, die zunächst eingeführt werden.

Die Referenzarchitektur unterscheidet zwischen den folgenden Architektursichten:

**Fachliche Referenzarchitektur:** Dies umfasst eine fachliche Beschreibung der Anwendungslandschaft. Diese wird in fachliche Komponenten zerlegt, die miteinander über fachliche Schnittstellen interagieren.

**Software-Referenzarchitektur:** Dies ist eine software-technische Beschreibung der Systemlandschaft. Die gesamte Systemlandschaft wird in technische Komponenten mit technischen Schnittstellen zerlegt.

**Referenzarchitektur der technischen Infrastruktur:** Hierunter fällt eine Beschreibung der Systemlandschaft aus dem Blickwinkel der technischen Infrastruktur. Es wird beschrieben, welche technischen

Systeme existieren und welche technischen Komponenten auf ihnen laufen.

Diese Architektursichten werden im Folgenden beschrieben.

### 1.1.1 Fachliche Referenzarchitektur

Eine Anwendungslandschaft baut sich aus Anwendungsdomänen, Anwendungssystemen (Fachanwendungen und Querschnittssystemen), dem Portal und den Service-Gateways auf. Dies ist in Abbildung 1 dargestellt.

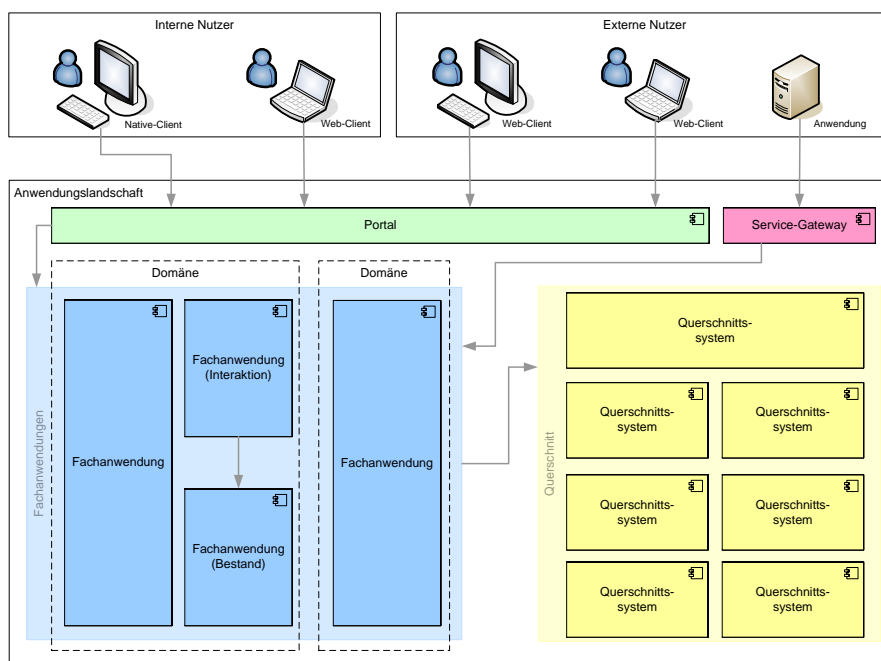


Abbildung 1: Fachliche Architektur einer Anwendungslandschaft

**Anwendungsdomänen:** Anwendungsdomänen sind fachlich eng zusammengehörige Mengen von Anwendungssystemen zur Unterstützung von Geschäftsprozessen, die als eine Einheit angesehen werden können.

**Anwendungssysteme:** Ein Anwendungssystem ist eine zusammengehörende, logische Einheit aus Funktionen, Daten und Benutzungsschnittstellen. Anwendungssysteme können monolithisch aufgebaut so dass sie nicht auf andere Anwendungssysteme angewiesen sind, um ihre Funktion zu erfüllen. Es ist jedoch auch möglich, bestimmte Funktionen und Schnittstellen über mehrere Anwendungssysteme zu verteilen, um so eine klare architektonische Strukturierung der Anwendungslandschaft zu erreichen. Die Gesamtfunktionalität wird dann durch das Zusammenwirken der Anwendungssysteme erbracht.

Ein häufig anzutreffendes Beispiel hierfür ist die Trennung in ein Anwendungssystem zur Bestandsverwaltung (Register) und ein Anwendungssystem zur Benutzerinteraktion, die auf die Bestandsverwaltung zugreift. Im Zuge einer solchen Aufgabenteilung

werden auch weitergehende Regeln definiert, wie Anwendungssysteme aufeinander zugreifen dürfen. Derartige Strukturierungsprinzipien und Regeln sind jedoch immer spezifisch für den jeweiligen Kontext der Anwendungslandschaft und können nicht allgemeingültig festgelegt werden. Im vorliegenden Dokument verwenden wir daher den neutralen Begriff **Fachanwendung** für alle Arten von Anwendungssystemen, die eine bestimmte fachliche Funktionalität innerhalb einer Anwendungsdomäne realisieren.

**Querschnittssysteme** sind Anwendungssysteme, die Dienste für mehrere Fachanwendungen bereitstellen. Diese Dienste sind oft technisch motiviert.

**Portal:** Das Portal stellt den Benutzern eine einheitliche Dialogoberfläche für den Zugriff auf die Dienste der Anwendungslandschaft zur Verfügung. Einzelne Fachanwendungen stellen Nutzerschnittstellen zur Verfügung, das Portal integriert diese, bildet einen zentralen Einstiegspunkt für alle Web-Anwendungen und übernimmt die Aufgabe der Authentifizierung und der Autorisierung auf Anwendungsebene.

**Service-Gateway:** Im Service-Gateway werden alle Services der Anwendungssysteme bereitgestellt, die externe Nutzer direkt nutzen dürfen. Dazu führt das Service-Gateway die Authentifizierung und Autorisierung der Service-Aufrufer auf Anwendungsebene durch. Weiterhin transformiert das Service-Gateway das interne Kommunikationsprotokoll einer IsyFact-Systemlandschaft in das vom Service-Aufrufer erwartete Protokoll. Dies sind in der Regel Web Service-Aufrufe.

### 1.1.2 Software Referenzarchitektur

Die individuell erstellten Anwendungssysteme der Anwendungslandschaft sind technisch gleichartig aufgebaut. Daher wird eine Referenzarchitektur aufgestellt, nach denen Anwendungssysteme (Fachanwendungen und Querschnittssysteme) software-technisch in IT-Systeme umgesetzt werden. Das Dokument [IsyFactReferenzarchitekturITSystem] beschreibt diese Referenzarchitektur.

Skalierbarkeit und Hochverfügbarkeit sind wichtige Anforderungen an die Referenzarchitektur. Dies wird unterstützt durch eine zustandslose Serverarchitektur. Der Zustand einer Anwendung wird in der Datenbank persistiert und bei jedem eintreffenden Service-Aufruf aus dieser gelesen und bei Beendigung des Service-Aufrufs wieder in die Datenbank geschrieben.



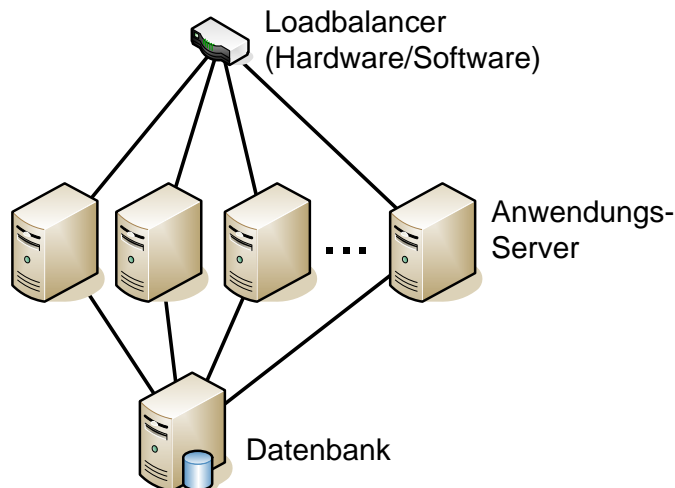


Abbildung 2: Horizontale Skalierung

In der Referenzarchitektur erfolgt die Anpassung an steigende Anforderungen durch horizontale Skalierung auf der Ebene der Anwendungsserver. Ein (Hardware- oder Software-) Loadbalancer verteilt die eingehenden Anfragen auf die vorhandenen Anwendungsserver. Im Falle eines Serverausfalls kann die Aufgabe des ausgefallenen Servers durch einen anderen Server übernommen werden. Die Skalierung des Systems ist in Abbildung 2 dargestellt.

Die Referenzarchitektur ist die Umsetzung einer Serviceorientierten Architektur. Im Design des Anwendungskerns (siehe [AnwendungskernDetailkonzept]) finden sich explizit Komponenten und Services. Der Anwendungskern unterscheidet dabei noch zwischen Anwendungs-internen Services, die nur innerhalb der Anwendung aufgerufen werden und Anwendungs-externen Services, die über eine Nutzungsschnittstelle als Service anderen Anwendungen zur Verfügung gestellt werden.

Der Entwurf der Services leitet sich in der Referenzarchitektur aus fachlichen Kriterien her. Services werden in der Fachlichen Referenzarchitektur identifiziert und finden sich dann auch in der technischen Implementierung wieder.

Die in diesem Dokument beschriebene Referenzarchitektur ist eine vollwertige JEE-Architektur. Jedoch wird eine zentrale Spezifikation von JEE nicht genutzt: die EJB-Spezifikation aus dem Bereich Enterprise Application. Dies hat vor allem Performance- und Komplexitätsgründe. Es hat zur Folge, dass als Application Server ein Servlet-Container ausreichend ist.

### 1.1.3 Referenzarchitektur der technischen Infrastruktur

Im Bereich der technischen Infrastruktur (TI-Architektur) werden folgende Umgebungen beschrieben:

- Produktionsumgebung
- Staging-Umgebung

- Schulungs- und externe Testumgebung
- Entwicklungs- und Abnahme-Testumgebung

Die Aufteilung in Zonen leitet sich aus dem SAGA 4-Standard<sup>1</sup> ab [SAGA40].

Abbildung 3 skizziert die Referenzarchitektur der technischen Infrastruktur für die Produktionsumgebung. Die anderen Umgebungen sind vereinfachte und verkleinerte Abbilder der Produktionsumgebung.

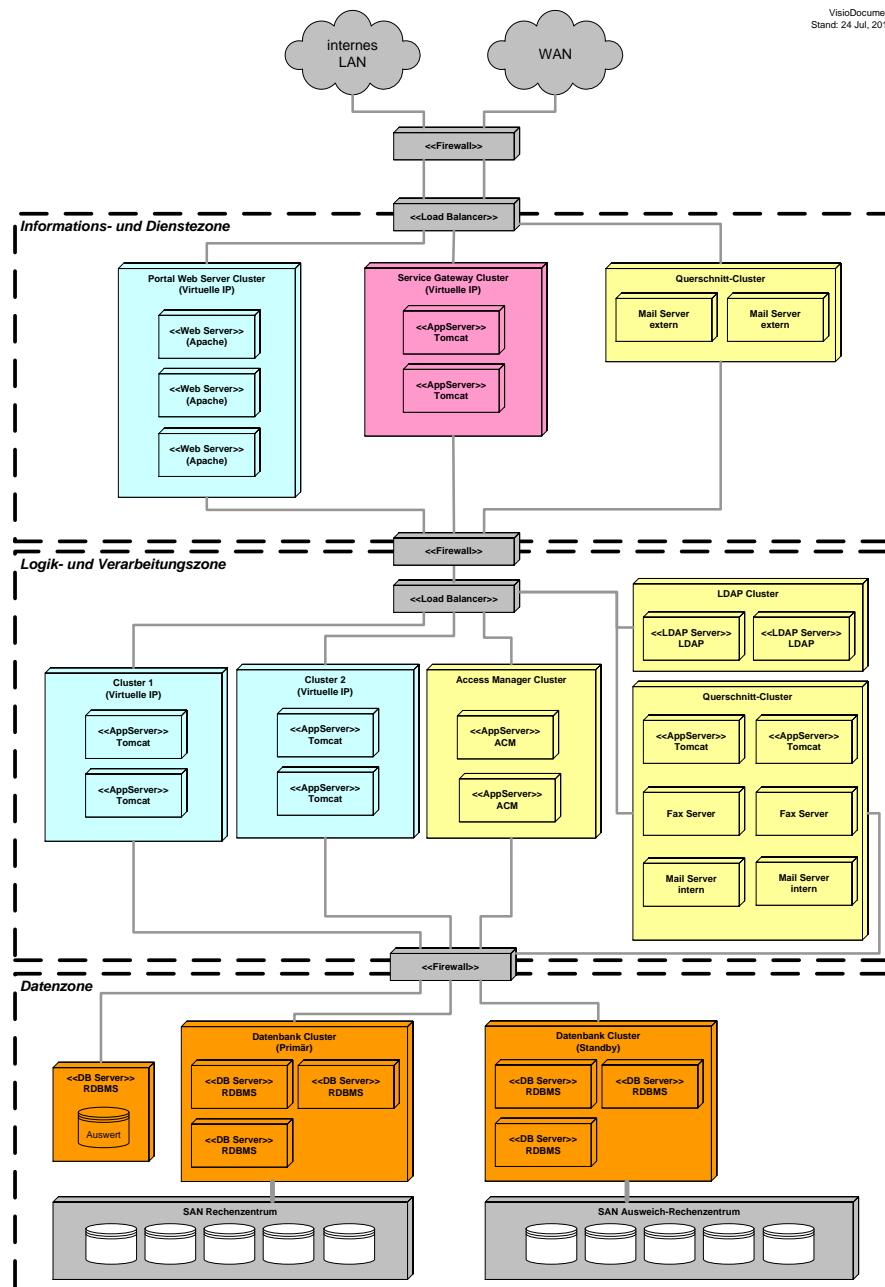


Abbildung 3: Referenzarchitektur der technischen Infrastruktur für die Produktionsumgebung

<sup>1</sup> Wir orientieren uns hier nach wie vor am SAGA 4-Standard, da SAGA 5 kein Zonenmodell mehr enthält.

Um die Sicherheit in der Datenkommunikation zu gewährleisten, sind die Server unterschiedlichen Sicherheitszonen des Netzwerks zugeordnet. In Abbildung 3 ist eine Sicherheitszone durch ein gestricheltes Rechteck dargestellt. Zonenübergreifende Kommunikationsverbindungen werden von den Firewalls kontrolliert.

Für die Datenhaltung wird ein auf einem relationalen Datenbank-Management-System (RDBMS) basierender Datenbank-Cluster im Hot-Standby eingesetzt. Um Auswertungen auf Stichtagsbeständen durchführen zu können, wird ein dedizierter Datenbankserver vorgesehen.

## **1.2. Inhaltsübersicht**

Nachdem in Kapitel 1.1 die drei Sichten der Referenzarchitektur für eine Anwendungslandschaft kurz vorgestellt wurden, erfolgt nun eine Detaillierung der einzelnen Sichten:

- Für die fachliche Sicht wird in Kapitel 3 eine fachliche Referenzarchitektur beschrieben.
- Für die softwaretechnische Sicht wird in Kapitel 4 die Einbindung der Software-Referenzarchitektur [IsyFactReferenzarchitektur|ITSystem] beschrieben.
- Für die technische Infrastruktur wird in Kapitel 5 eine Referenzarchitektur vorgestellt

Im Rahmen dieser Referenzarchitektur werden auch betriebliche Aspekte betrachtet, da diese Auswirkungen auf Designentscheidungen haben können. Diese Aspekte werden in Kapitel 6 beschrieben.

## 2. Übersicht

Dieses Kapitel führt in die grundlegenden Ideen der IsyFact ein. Dazu wird zunächst die IsyFact selbst vorgestellt, danach die Referenzarchitektur in die IsyFact eingeordnet und Ideen und Ziele der Referenzarchitektur beschrieben.

### 2.1. Die IsyFact

Dieses Dokument und die enthaltene Referenzarchitektur sind Bestandteile der IsyFact. Im Folgenden wird beschrieben, was die Idee hinter der IsyFact ist und welche inhaltlichen Bereiche die IsyFact insgesamt abdeckt.

#### 2.1.1 Die Idee der IsyFact

Beschäftigt man sich mit dem Entwurf und der Implementierung von Anwendungslandschaften, so stellt man fest, dass sich die einzelnen Fachanwendungen im Allgemeinen sehr ähnlich sind. Viele Problemstellungen müssen nur einmal gelöst werden und können dann wieder verwendet werden.

Die IsyFact wurde entwickelt, um unterschiedliche Fachanwendungen effizient entwerfen, realisieren und betreiben zu können. Sie umfasst Konzepte, Software-Bausteine, eine Betriebsplattform, Werkzeuge und eine Methodik, auf deren Basis Fachanwendungen schneller, preiswerter und sicherer umgesetzt werden können. Der Begriff Factory steht für die schnelle Erzeugung neuer Anwendungen durch Anpassung und Ausgestaltung vorgefertigter Elemente, die in eine standardisierte Plattform integriert werden können.

Die Vorteile des IsyFact-Ansatzes sind vielfältig: Die zügige und effiziente Einrichtung neuer Fachanwendungen erhöht die Wirtschaftlichkeit. Durch Homogenisierung der Anwendungslandschaft werden Administration und Wartung vereinfacht. Weitere Vorteile sind höhere Sicherheit und Softwarequalität durch Einsatz erprobter Komponenten und Dienste sowie mehr Nachhaltigkeit durch eine zukunftssichere, weitgehend herstellernerneutrale Software-Architektur und Plattform.

## 2.1.2 Durch die IsyFact abgedeckte Bereiche

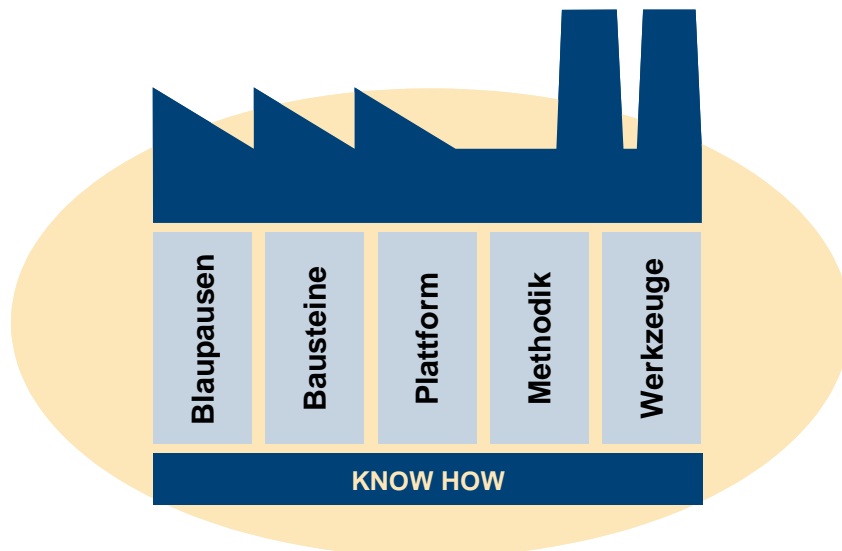


Abbildung 4: Durch die IsyFact abgedeckte Bereiche

Die IsyFact bietet standardisierte Lösungsbausteine zu wichtigen Themen beim Bau von Fachanwendungen. Der Entwickler kann sich dadurch ganz auf die strukturellen und operativen Besonderheiten seiner Fachanwendung konzentrieren.

Die Lösungsbausteine decken fünf Bereiche ab: Blaupausen, Bausteine, Betriebsplattform, Methodik und Werkzeuge.

**Blaupausen:** Blaupausen beschreiben die Architektur und Konzepte einer Anwendungslandschaft für den Betrieb von Fachanwendungen aus drei verschiedenen Sichten: der fachlichen Sicht, der Sicht der softwaretechnischen Umsetzung und der Sicht der technischen Infrastruktur.

**Bausteine:** Bausteine sind konkrete Lösungen zur Umsetzung der in den Blaupausen beschriebenen Architektur und Konzepte. Bausteine sind wieder verwendbare Softwarelösungen, z. B. ein Regelwerk für fachliche Konsistenzprüfungen oder Komponenten zur Protokollierung. Diese Bausteine liegen in unterschiedlichen Formen vor: Es gibt fachliche und technische Services im Sinne einer serviceorientierten Architektur (SOA) und wieder verwendbare Bibliotheken und Programmiervorlagen. Dazu gehören auch am Markt verfügbare Fertigprodukte, für die detaillierte Nutzungsvorgaben erstellt wurden.

**Betriebsplattform:** Die Betriebsplattform besteht aus Hardware und Netzen sowie der benötigten Middleware mit Anwendungsserver oder Datenbankserver und ist hochverfügbar. Diese Plattform ist über alle Fachanwendungen einheitlich und ermöglicht so einen standardisierten und effizienten Systembetrieb.

**Methodik:** Die Methodik der IsyFact bietet verschiedene Vorschläge zu allen Phasen des Software-Entwicklungsprozesses, wie z. B. Vorlagen für die Spezifikation oder die Programmierung. Das Projektvorgehen selbst ist V-Modell®-XT-konform.

**Werkzeuge:** Die IsyFact setzt auf Automatisierung und Werkzeugunterstützung bei der Erstellung von Registern. Dazu bietet sie vorkonfigurierte Werkzeuge, z. B. für die Modellierung, die Programmierung, Tests oder die Fehlerverfolgung.

## 2.2. Einordnung der Referenzarchitektur in die IsyFact

Die in diesem Dokument beschriebene Referenzarchitektur ist ein zentraler Teil der IsyFact und erläutert auf einer hohen Ebene, wie eine Anwendungslandschaft zu beschreiben und umzusetzen ist. Die weiteren *Blaupausen* der IsyFact detaillieren Teile dieser Referenzarchitektur und werden im vorliegenden Dokument referenziert.

## 2.3. Die Idee der Referenzarchitektur

Die Referenzarchitektur definiert einen Strukturierungsrahmen für die Anwendungslandschaft und legt Prinzipien für die softwaretechnische Realisierung fest.

Der konkrete Aufbau einer Anwendungslandschaft mit der IsyFact orientiert sich an dieser Referenzarchitektur. Dabei wird die Referenzarchitektur als Ausgangspunkt genommen und an den Stellen modifiziert und konkretisiert, die im konkreten Kontext nicht direkt anwendbar sind.

Im Rahmen der Referenzarchitektur wird eine solche Anwendungslandschaft aus der fachlichen Sicht, der Sicht der softwaretechnischen Umsetzung und der Sicht der technischen Infrastruktur betrachtet. Für jede Sicht wird eine Referenzarchitektur definiert:

**Fachliche Referenzarchitektur:** Hierunter fällt eine fachliche Beschreibung der Anwendungslandschaft. Die gesamte Anwendungslandschaft wird in fachliche Komponenten zerlegt, die miteinander über fachliche Schnittstellen interagieren. Jegliche technische Aspekte sind dabei nicht von Belang. Die fachliche Referenzarchitektur wird in Kapitel 3 definiert.

**Software-Referenzarchitektur:** Hierunter fällt eine software-technische Beschreibung der Systemlandschaft. Die gesamte Systemlandschaft wird in technische Komponenten zerlegt, die miteinander über technische Schnittstellen interagieren. Jegliche Abbildung auf physikalische Systeme und deren Infrastruktur sind dabei nicht von Belang. Die Software-Referenzarchitektur wird in [IsyFactReferenzarchitektur|TSystem] definiert.

**Referenzarchitektur der technischen Infrastruktur:** Ist eine Beschreibung der Systemlandschaft aus dem Blickwinkel der technischen Infrastruktur. Es wird beschrieben, welche technischen Systeme existieren und welche technischen Komponenten auf ihnen laufen. Die Referenzarchitektur der technischen Infrastruktur wird in Kapitel 5 definiert.

## 2.4. Die Ziele der Referenzarchitektur

Mit der Einführung und Verwendung der Referenzarchitektur werden unter anderem die folgenden Ziele verfolgt:

**Einheitlichkeit:** Durch die Vorgaben der Referenzarchitektur wird eine Gleichartigkeit der verschiedenen Systeme in Hinblick auf ihre fachliche Strukturierung, ihre technologische Strukturierung und Umsetzung, die verwendeten Produkte, ihre Dokumentation und die Art ihres Betriebs erreicht. Hierdurch verbessert sich die Verständlichkeit, Wartbarkeit und Betreibbarkeit der Systeme.

**Effizienz:** Durch die Verwendung vorhandener Querschnittssysteme, Bibliotheken, Generatoren, Beispielanwendungen und die Vorgaben der Referenzarchitektur können die Anwendungssysteme effizienter entwickelt und gewartet werden. Durch die infrastrukturelle und betriebliche Gleichartigkeit der Anwendungssysteme können diese effizienter betrieben werden.

**Sicherheit:** Die Referenzarchitektur sieht bewährte Technologien und Produkte, eine SAGA-konforme Infrastruktur, und nach klaren architektonischen Vorgaben erstellte Anwendungssysteme vor. Die Risiken für Ausfälle, Fehler oder Sicherheitslücken wird hierdurch reduziert und die Zukunftssicherheit der Anwendungssysteme erhöht.

**Erfahrung:** In die Definition der Referenzarchitektur sind wertvolle Erfahrungen zur Umsetzung großer und verteilter Softwaresysteme im Allgemeinen und Registern im Speziellen eingeflossen. Diverse Fachanwendungen wurden mit der Referenzarchitektur umgesetzt, und die Referenzarchitektur kontinuierlich verbessert. Durch die Verwendung der Referenzarchitektur wird diese Erfahrung nutzbar gemacht. Die mit dieser Referenzarchitektur erstellten Fachanwendungen sind hierdurch äußerst effizient umsetzbar, wartbar, betreibbar und zukunftssicher.

**Standards:** Durch die Referenzarchitektur werden Standards wie SAGA, V-Modell XT, XML / XÖV oder Webservices genutzt.

## 2.5. Tailoring der Referenzarchitektur

Die IsyFact-Referenzarchitektur hat das Ziel, eine möglichst große Bandbreite von Systemen abzudecken. Daher trifft sie zunächst kaum Annahmen über die Struktur der Systemlandschaft oder die Art der darin enthaltenen Systeme. Auf einer allgemeinen Ebene unterscheidet die IsyFact-Referenzarchitektur lediglich zwischen Fachanwendungen, welche

die Geschäftsprozesse innerhalb der Systemlandschaft implementieren einerseits, und andererseits den Querschnittssystemen, Service-Gateways und dem Portal, die als unterstützende Systeme wesentliche Strukturkonzepte der IsyFact umsetzen.

In einem spezifischen Anwendungskontext wird man in der Regel jedoch viel weitergehende Festlegungen innerhalb der Anwendungs-Architektur treffen, die die Fachlichkeit des Kontexts abbilden. Diese kontextspezifischen Festlegungen werden in einem eigenen *Tailoring-Dokument* [TailoringVorlage] festgehalten, das beschreibt, wie die IsyFact im betreffenden Anwendungskontext einzusetzen ist. Insbesondere enthält das Tailoring-Dokument die Arten von Fachanwendungen und ihre Beziehungen untereinander.

Gemeinsam mit dem vorliegenden Dokument definiert das Tailoring Dokument die Referenzarchitektur für eine spezifische Fachlichkeit in einem Anwendungskontext (Kontext-Fachlichkeit). Dadurch ist es möglich, die Software-Architektur und die Architektur der technischen Infrastruktur in unterschiedlichen Kontexten zu verwenden und auf eine spezifische Fachlichkeit anzupassen. Dies ist in Abbildung 15 dargestellt.

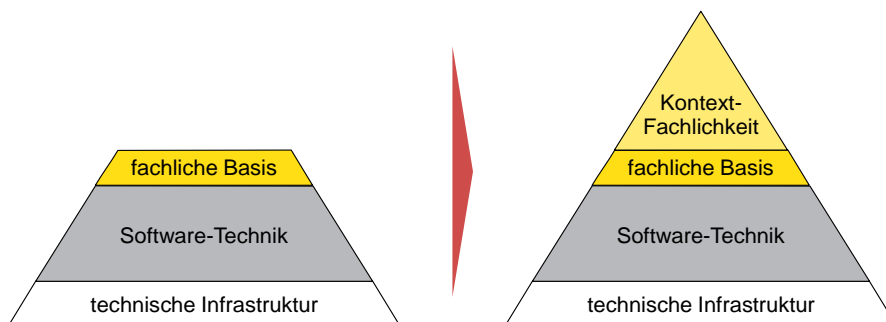


Abbildung 5: Anpassung der technischen Referenzarchitekturen auf den fachlichen Kontext



### 3. Die fachliche Referenzarchitektur

Zu den Aufgaben einer öffentlichen oder privatwirtschaftlichen Organisation gehört die Durchführung verschiedener fachlicher Verfahren. Wenn solche Verfahren vollständig oder teilweise automatisiert werden sollen, so erfolgt dies in der Regel mit Anwendungssystemen, die zunächst aus fachlicher Sicht beschrieben, dann mit softwaretechnischen Mitteln umgesetzt und schließlich auf einer technischen Infrastruktur betrieben werden.

Bei einer großen Zahl von vollständig oder teilweise automatisierten Verfahren entsteht eine entsprechende Anzahl von Anwendungssystemen. Hier kann schnell der Überblick verloren gehen, so dass Prinzipien zur Strukturierung benötigt werden, die Ordnung schaffen.

Die Menge aller Anwendungssysteme einer Organisation und deren Nutzungsbeziehungen untereinander bilden eine Anwendungslandschaft. Die fachliche Architektur einer Anwendungslandschaft gibt hier die Strukturierung aus fachlicher Sicht vor und legt fest, wie Anwendungssysteme in die Anwendungslandschaft integriert werden.

Die fachliche Architektur einer Anwendungslandschaft ist in Abbildung 6 dargestellt.

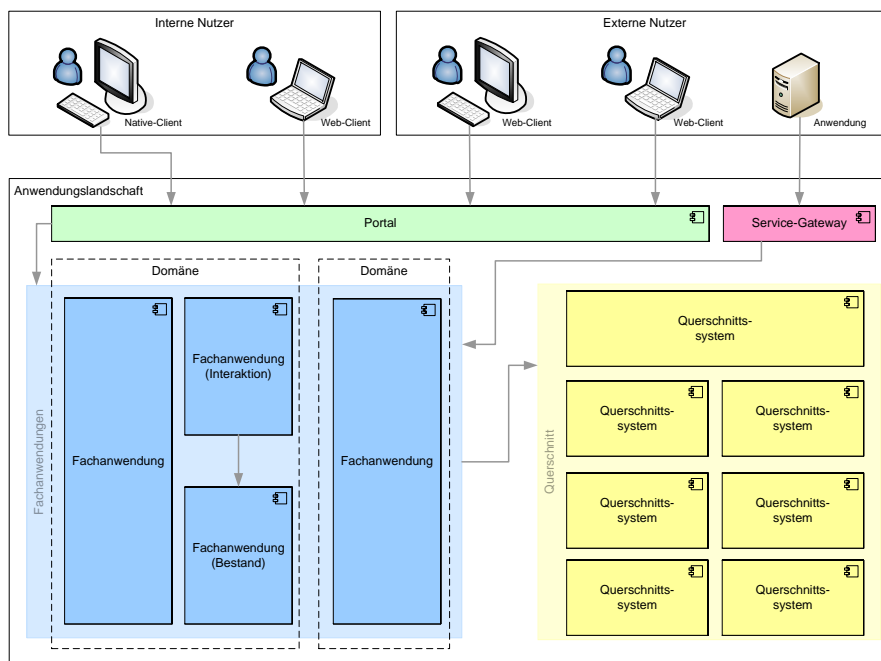


Abbildung 6: Fachliche Architektur einer Anwendungslandschaft

#### 3.1. Die Strukturierung der Anwendungslandschaft

Zur Strukturierung der Anwendungslandschaft werden drei fachliche Hierarchieebenen festgelegt:

**Anwendungsdomänen:** Auf der obersten Hierarchieebene werden Anwendungsdomänen gebildet: Anwendungsdomänen sind fachlich eng zusammengehörige Mengen von Anwendungssystemen zur

Unterstützung von Geschäftsprozessen, die als eine Einheit angesehen werden können. In Abbildung 6 wurden zwei Domänen gebildet, erkennbar an den unteren schwarzen Kästen mit der Beschriftung „Domäne“.

Zwischen den Anwendungssystemen unterschiedlicher Anwendungsdomänen sollten nur wenige, klar definierte Nutzungsbeziehungen existieren.

**Anwendungssysteme:** Ein Anwendungssystem ist eine zusammengehörende, logische Einheit aus Funktionen, Daten und Benutzungsschnittstellen. Geschäftsprozesse werden durch Anwendungssysteme unterstützt. Anwendungssysteme sind einer Anwendungsdomäne zugeordnet. Anwendungssysteme setzen sich aus Anwendungskomponenten zusammen. Im Rahmen der allgemeinen Referenzarchitektur werden zunächst nur Fachanwendungen und Querschnittssysteme als Typen von Anwendungssystemen unterschieden. Eine weitergehende Unterscheidung der Fachanwendungen muss im Rahmen des Tailorings der Referenzarchitektur erfolgen.

**Fachanwendungen:** Eine Fachanwendung implementiert für sich genommen oder im Zusammenspiel mit anderen Fachanwendungen einen oder mehrere Geschäftsprozesse einer Anwendungsdomäne. Sie implementiert entweder die gesamte hierfür notwendige Funktionalität (monolithisch), von der Benutzerschnittstelle über die fachliche Logik, die Prozesse bis hin zur Datenhaltung. Oder die Fachanwendung implementiert nur einen Teilbereich der Funktionalität und greift für den Rest über Schnittstellen auf benachbarte Fachanwendungen zu.

**Querschnittssysteme** sind Anwendungssysteme, die Dienste für mehrere Fachanwendungen bereitstellen, wie z. B. ein Behördenverzeichnis oder ein Outputmanagement.

**Anwendungskomponenten:** Eine Anwendungskomponente beschreibt eine Menge funktional zusammenhängender Anwendungsfälle. Anwendungskomponenten sind Bestandteile von Anwendungssystemen. Im Rahmen weitergehender Architekturvorgaben beim Einsatz der IsyFact in einem konkreten Anwendungskontext wird man in der Regel auch vorgeben, welche Arten von Anwendungssystemen es in diesem Kontext geben soll und aus welchen Komponenten sie bestehen.

In Abbildung 6 werden Anwendungskomponenten nicht dargestellt: Sie wären die Bestandteile der blauen und gelben Kästen.

### 3.2. Fachliche Referenzarchitekturen für Anwendungssysteme

Die fachliche Referenzarchitektur muss beim Einsatz der IsyFact innerhalb des jeweiligen Anwendungskontexts definiert werden. Die IsyFact dient hier

als allgemeiner technischer und konzeptioneller Rahmen und macht keine Vorgaben über die Arten von Anwendungssystemen, die in einem bestimmten Kontext vorkommen.

Als Teil der fachlichen Architektur ist unter anderem folgendes zu definieren:

- Die verschiedenen Arten von Fachanwendungen, die es innerhalb des Anwendungskontexts geben soll.
- Die Aufgaben und Verantwortlichkeiten der einzelnen Arten von Fachanwendungen.
- Der daraus resultierende interne Aufbau der Fachanwendungen aus Komponenten.
- Die Interaktion und die Abhängigkeiten zwischen den Fachanwendungen, insbesondere die zulässigen Kommunikations-Beziehungen.

## 4. Die Software-Referenzarchitektur

Eine Software-Architektur beschreibt, wie die in einer fachlichen Architektur definierten Elemente (Anwendungssysteme, Anwendungskomponenten, fachliche Entitäten, Anwendungsfälle etc.) software-technisch in Form von IT-Systemen, Komponenten, Klassen, physischen Datenmodellen etc. umgesetzt werden.

Die Besonderheit der Software-Referenzarchitektur ist, dass sie nicht zwischen Systemarten (Fachanwendungen, Querschnittssystemen, usw.) unterscheidet. Obwohl unterschiedliche Systeme sehr unterschiedliche Fachlichkeiten umsetzen können, sind die Anforderungen an ihre technische Architektur gleich.

Der Begriff der Anwendungslandschaft ist fachlich motiviert. Die technische Entsprechung hierfür ist der Begriff der Systemlandschaft. Eine Systemlandschaft, die eine Anwendungslandschaft nach den IsyFact-Standards umsetzt, wird im Folgenden als *IsyFact-Systemlandschaft* bezeichnet. Eine IsyFact-Systemlandschaft beinhaltet alle software-technisch umgesetzten Anwendungssysteme der Anwendungslandschaft sowie technische Systeme zur Unterstützung (z. B. Datenbanken, Web-Server, usw.). Sie wird auf einer technischen Infrastruktur betrieben und bildet einen in sich geschlossenen Sicherheitsbereich.

Die technische Architektur eines IT-Systems wird in [IsyFactReferenzarchitektur|ITSystem] definiert. Im Folgenden wird, aufbauend auf diesem Konzept, beschrieben

- welche Eigenschaften bezogen auf die technische Referenzarchitektur die Typen von Anwendungssystemen in einer IsyFact-Systemlandschaft besitzen (Kapitel 4.1),
- welche Vorgaben für die Verwendung von Produkten existieren (Kapitel 4.3).

### 4.1. Strukturierung der Register-Plattform

Wichtige Grundlagen für die Software-Referenzarchitektur sind die Schnittstellen und Aufruf-Beziehungen der IT-Systeme vom Typ *Fachanwendung*, *Querschnittssystem* sowie *Portal* und *Service-Gateway*. Aufruf-Beziehungen werden stets unterschieden in *interne Aufrufe*, in welchen ein IT-System einer IsyFact-Systemlandschaft mit einem anderen IT-System derselben IsyFact-Systemlandschaft kommuniziert, und *externe Aufrufe*, in welchen ein IT-System der IsyFact-Systemlandschaft mit einem Anwender oder einem IT-System außerhalb dieser IsyFact-Systemlandschaft kommuniziert: Interne und externe Aufrufe unterscheiden sich sowohl in ihrer Authentifizierung und Autorisierung als auch (bei automatisierten Schnittstellen) in der verwendeten Technologie.

Im Folgenden werden die Schnittstellen und Aufrufbeziehungen pro Anwendungssystemtyp erläutert.

**Das Service-Gateway:** Service-Gateway-Systeme haben die Aufgabe, Aufrufe von internen Anwendungssystemen an externe Systeme und Aufrufe von externen Systemen an interne Anwendungssysteme weiterzuleiten und dabei unterschiedliche Schnittstellentechnologien zu überbrücken.

Für die Weiterleitung von Aufrufen enthält ein Service-Gateway die folgenden Funktionalitäten:

- Es bildet die Aufrufe zwischen der für externe Systeme verwendeten Webservice-Technologie und der für interne Schnittstellen verwendeten Http-Invoker-Technologie aufeinander ab. Es kapselt die Webservice-Technologie vor den internen Systemen.
- Bei Aufrufen durch externe Systeme führt es die Authentifizierung und eine erste Autorisierung des Aufrufs durch.

Ein Service-Gateway-System besitzt weder eine eigene Datenhaltung noch eigene Fachlichkeit.

**Die Fachanwendung:** Eine Fachanwendung implementiert allgemein die Geschäftsprozesse einer Domäne. Fachanwendungen können monolithisch strukturiert sein und alle Schichten der Software-Architektur umfassen, von der Benutzeroberfläche über Prozesse und Geschäftslogik bis hin zur Datenspeicherung. Sie können aber auch jeweils nur Teile davon implementieren, so dass z.B. eine Fachanwendung eine GUI bereitstellt, eine weitere die Fachlogik implementiert und eine dritte schließlich die Persistierung der Daten realisiert. In diesem Fall würde die Gesamtfunktionalität durch das Zusammenwirken der drei Fachanwendungen erbracht.

Die Fachanwendungen einer IsyFact-Systemlandschaft können sich intern gegenseitig über ihre Service-Schnittstellen aufrufen. In einem konkreten Anwendungskontext einer IsyFact-Systemlandschaft wird man in der Regel die Arten von Fachanwendungen noch genauer definieren und dabei auch die möglichen Aufrufbeziehungen zwischen ihnen geeignet einschränken.

Für die Kommunikation mit externen Systemen muss eine Fachanwendung ein Service-Gateway-System verwenden, egal in welche Richtung die Kommunikation verläuft.

Fachanwendungen, die eine Benutzeroberfläche bereitstellen, dürfen aus Sicherheitsgründen nicht direkt vom Benutzer aufgerufen werden, sondern werden in das *Portal* (siehe unten) der IsyFact-Systemlandschaft eingebunden. Die Kommunikation mit dem Browser des Nutzers muss immer über das Portal erfolgen.

**Das Portal:** Die Referenzarchitektur sieht keinen Portal-Server im klassischen Sinne vor: Es gibt weder einen systemübergreifenden Rahmen noch eine systemübergreifende Navigation.

Ein Portal im Sinne der Referenzarchitektur besteht aus:

- Einem Web-Server-Cluster, der Aufrufe entgegennimmt und an die Application-Server der Fachanwendungen weiterleitet.
- Einem Loadbalancer, der die Lastverteilung im Web-Server-Cluster vornimmt und die SSL-Verbindung terminiert.
- Einem Access-Manager, über den ein Single-Sign-On realisiert wird und in dem pro URL (und damit pro Fachanwendung) konfiguriert ist, welche Berechtigungen für die Nutzung der Anwendung vorhanden sein müssen. Innerhalb dieses Systems werden die Authentifizierung, die Prüfung der Sitzungszertifikate und eine grundlegende Autorisierung vorgenommen.
- Einer Startseite, welche den eingeloggten Benutzern die Fachanwendungen der Register-Plattform, für welche sie berechtigt sind, präsentiert.

Ein Portal wird somit nicht über ein eigens dafür implementiertes Programm, sondern über konfigurierte Software-Produkte umgesetzt. Der Zweck eines Portals ist die gemeinsame Authentifizierung und Autorisierung für alle Fachanwendungen und die Indirektion des Zugriffs von Nutzern auf die Geschäftsanwendungen.

**Das Querschnittssystem:** Ein Querschnittssystem bietet Fachanwendungen querschnittlich genutzte Funktionalitäten an, etwa für die Bereitstellung von Schlüsseldaten. Querschnittssysteme können eine eigene Datenhaltung besitzen.

Querschnittssysteme werden nicht direkt von Benutzern oder externen Systemen außerhalb der IsyFact-Systemlandschaft aufgerufen und rufen selbst nicht direkt externe Systeme auf. Ausnahmen hiervon sind Querschnittssysteme wie das Outputmanagement das z. B. zur Versendung von externen E-Mails einen Mailserver nutzt oder zur Versendung von Faxen einen Fax-Server.

## 4.2. Nutzungsarten eines Anwendungssystems

Nachdem im [IsyFactReferenzarchitekturITSystem] die technische Architektur vorgestellt wurde, soll in diesem Abschnitt konkret vorgestellt werden, wie auf Basis dieser Architektur Anwendungen entworfen werden.

Die Nutzungsschicht eines IT-Systems bietet anderen IT-Systemen über Services, dem Betrieb über Batches und menschlichen Nutzern über eine GUI Schnittstellen zur Nutzung der implementierten Fachlichkeit an. Im Folgenden wird ein Beispiel-Szenario zur Nutzung eines IT-Systems vorgestellt.

Es soll eine Fachanwendung erstellt werden, die sowohl manuell über eine Weboberfläche als auch automatisiert über eine SOAP-Schnittstelle nutzbar sein soll.

Für eine architekturkonforme Umsetzung dieser Anforderungen müssen verschiedene IT-Systeme umgesetzt werden:

**Die Fachanwendung:** Die Fachanwendung soll in diesem Beispiel die Präsentationslogik, die Geschäftslogik und die Datenhaltung in einem Anwendungssystem realisieren. Sie implementiert dazu drei software-architektonische Schichten:

- Die **Präsentationsschicht**, in der die Webseiten der GUI erzeugt und die Eingaben interpretiert werden.
- Der **Anwendungskern**, in dem die Geschäftslogik abgebildet ist.
- Die **Persistenzschicht**, in der sowohl die Anwendungsdaten als auch der Zustand der Anwendungssitzung abgespeichert werden.

Zur Implementierung der Geschäftslogik kann die Fachanwendung auch Services anderer Fachanwendung aufrufen.

**Das Portal:** Wie in Kapitel 4.1 beschrieben, nimmt das Portal die Benutzer-Aufrufe entgegen und leitet sie an die Geschäftsanwendung weiter.

**Das Service-Gateway-System:** Wie in Kapitel 4.1 beschrieben, dient das Service-Gateway-System als Schnittstelle für die Kommunikation mit bzw. der Annahme der Aufrufe von externen Systemen. Im vorliegenden Fall wird nur ein Service-Provider benötigt, da nur Aufrufe entgegengenommen werden. Diese werden authentifiziert, autorisiert, und an das zugehörige Anwendungssystem weitergeleitet.

Damit ergeben sich die in Abbildung 7 dargestellten Anwendungssysteme und Aufruf-Beziehungen. Die in dieser Abbildung angedeuteten Schichten eines Anwendungssystems (GUI, Batch, Service, Anwendungskern, Datenzugriff und Querschnitt) werden im Dokument [IsyFactReferenzarchitekturITSystem] erläutert. Wichtig in Hinblick auf diese Schichten sind folgende Punkte:

- Die Schicht *Batch* der Fachanwendung wird im Beispiel nicht implementiert, da die Fachanwendung keine Schnittstelle für Batchläufe anbietet.
- Alle externen Aufrufe an die Fachanwendung werden durch die Schicht *Service* verarbeitet: Keine andere Schicht darf externe Schnittstellen bereitstellen, insbesondere nicht die Schicht *Anwendungslogik*.

- Es wurden keine Aufrufe von Querschnittskomponenten eingezeichnet. Für derartige Aufrufe gibt es keine Vorgaben: Sie können aus beliebigen Schichten von Fachanwendungen und Service Gateways aus aufgerufen werden.

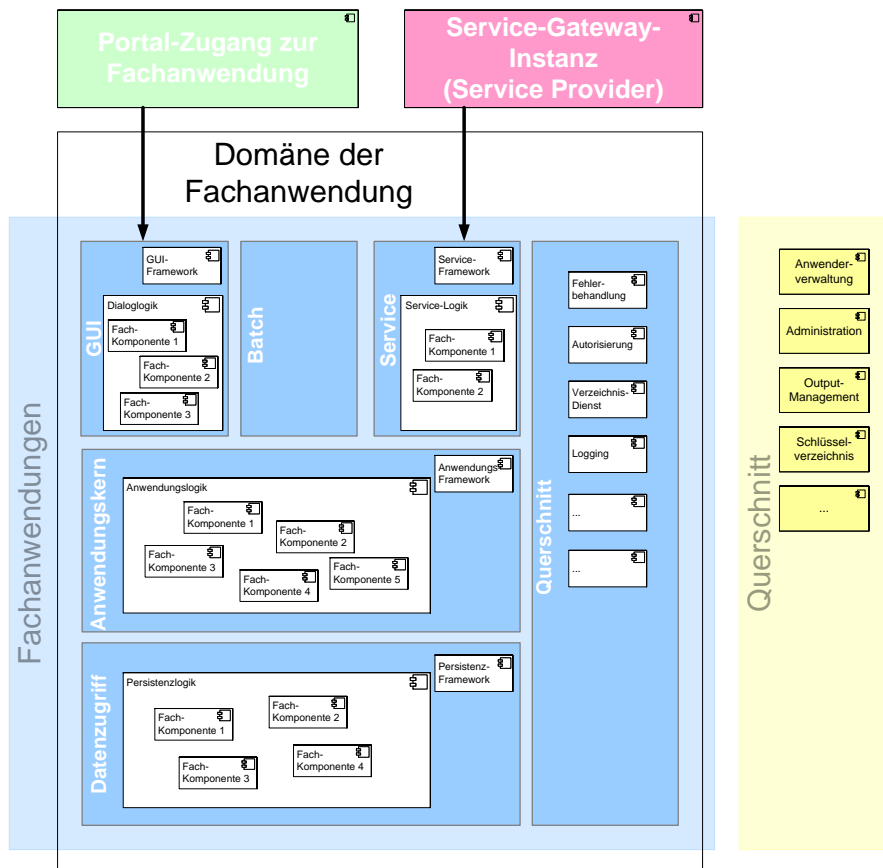


Abbildung 7: Aufrufbeziehungen für eine Fachanwendung mit GUI und XML-Schnittstelle

#### 4.3. Verwendung von Produkten

Bei der Umsetzung einer Architektur für eine Anwendung oder eine Anwendungslandschaft können an vielen Stellen Produkte eingesetzt werden. Das beschleunigt die Entwicklung und reduziert die Kosten.

Bei der Produktentscheidung sind zwei Seiten zu berücksichtigen: Auf der einen Seite bietet die Konzentration auf projektübergreifend einheitliche Produkte die Möglichkeit, die Fähigkeiten der Mitarbeiter zu bündeln und diese übergreifend einzusetzen. Auf der anderen Seite besteht die Gefahr, durch einen zu engen Fokus die Möglichkeiten eines Projekts zu sehr zu beschränken. Eine Lösung kann dann auch Gefahr laufen, zu allgemein zu werden, was letztlich die Komplexität steigert und größeren Aufwand verursacht.

Die für die Umsetzung der Architektur verwendeten Produkte lassen sich in die Kategorien Basistechnologien, Systemsoftware und Bibliotheken für die Anwendungsentwicklung unterteilen.



**Basistechnologien:** Basistechnologien legen grundlegende technische Entscheidungen fest, wie z.B. die Programmiersprache und die verwendete Web-Technologie.

**Systemsoftware:** Die Systemsoftware legt die technische Ablaufumgebung für die Software fest und bietet grundlegende Dienste für eine IsyFact-Systemlandschaft an. Hierzu gehören z.B. das Betriebssystem, Web-Server, Application-Server, Access-Manager, LDAP und die Datenbank.

**Bibliotheken für die Anwendungsentwicklung:** Die Anwendungsentwicklung wird durch den Einsatz von Frameworks und entsprechenden Bibliotheken vereinfacht und beschleunigt. IsyFact verwendet insbesondere Spring, Hibernate und JSF.

Eine detaillierte Liste der verwendbaren Produkte sind im Dokument [ProduktKatalog] zu finden.

## 5. Die Referenzarchitektur der technischen Infrastruktur

Die Referenzarchitektur der technischen Infrastruktur, auch TI-Architektur genannt, beschreibt den Aufbau der Betriebsumgebung für die IT-Systeme einer IsyFact-Systemlandschaft. Dazu gehören die physischen Geräte (Rechnersysteme, Netzwerkverbindungen und -komponenten, Drucker etc.), die installierte Systemsoftware (Betriebssystem, Applikationsserver, Middleware, Datenbanksystem) und das Zusammenspiel von Hardware und Systemsoftware.

Auf der Ebene der technischen Infrastruktur können mehrere Instanzen einer Komponente aus der technischen Architektur betrieben werden. Auch können mehrere Komponenten auf einem gemeinsamen Rechnersystem laufen.

### 5.1. Umgebungen

Um neben dem operativen Betrieb einer IsyFact-Systemlandschaft parallel neue Versionen von Anwendungen entwickeln, testen und schulen zu können, sind mehrere Systemumgebungen notwendig, die in Abbildung 8 vereinfacht dargestellt sind.

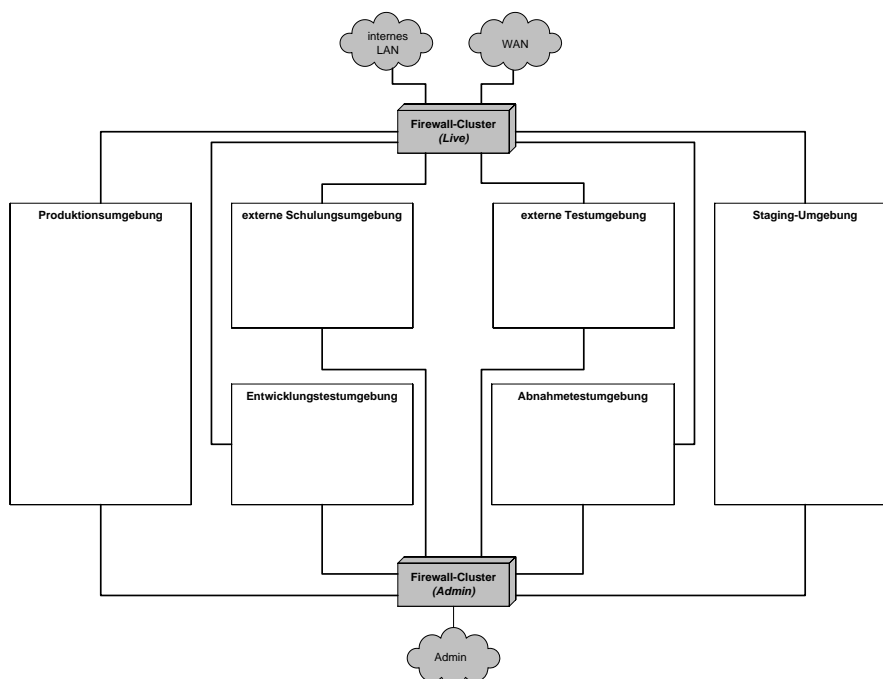


Abbildung 8: Überblick Systemumgebungen

Wir unterscheiden sechs Systemumgebungen:

- Produktionsumgebung
- Staging-Umgebung
- externe Schulungsumgebung
- externe Testumgebung
- Abnahmetestumgebung

- Entwicklungstestumgebung

Von internen Arbeitsplätzen sind prinzipiell alle Umgebungen erreichbar, sofern entsprechende Zugangsberechtigungen existieren. Die Administrationsarbeitsplätze befinden sich im Admin-Netz, von dem ebenfalls zu Administrationszwecken auf alle Systemumgebungen zugegriffen werden kann. Externe Anwender können nur bei entsprechender Berechtigung auf die Produktionsumgebung, die Schulungsumgebung und die externe Testumgebung zugreifen. Der Zugriff auf die Staging- sowie auf die Abnahmetestumgebung sowie die Entwicklungstestumgebung ist von Extern nicht zugelassen.

Die technischen Aspekte der gesamten Systemumgebungen werden nachfolgend erläutert. Für eine bessere Übersichtlichkeit in den Abbildungen der einzelnen Systemumgebungen, werden die Verbindungen mit dem Admin-Netz nicht dargestellt.

### **5.1.1 Produktionsumgebung (PRU)**

Mit dem Begriff „Produktionsumgebung“ wird die technische Infrastruktur bezeichnet, auf der der Wirkbetrieb einer IsyFact-Systemlandschaft abläuft. Alle nichtfunktionalen Anforderungen müssen von dieser Systemumgebung vollständig erfüllt werden.

Abbildung 9 skizziert die TI-Architektur der Produktionsumgebung. Einzelne Server werden durch UML-Knoten (engl. UML-Nodes) dargestellt. Die größeren Knoten gruppieren einzelne Server zu einer logischen Einheit, die *Cluster* genannt wird. Die Knoten eines Clusters sind dabei auf das Rechenzentrum und das Ausweich-Rechenzentrum verteilt, um bestmögliche Ausfallsicherheit zu erreichen. Die Verbindungen zeigen die Kommunikation der Server untereinander. Der Datenfluss erfolgt in der Regel in beiden Richtungen. Geht eine Verbindung von beziehungsweise zu einem Cluster, so entspricht das Verbindungen von beziehungsweise zu allen Servern im Cluster.

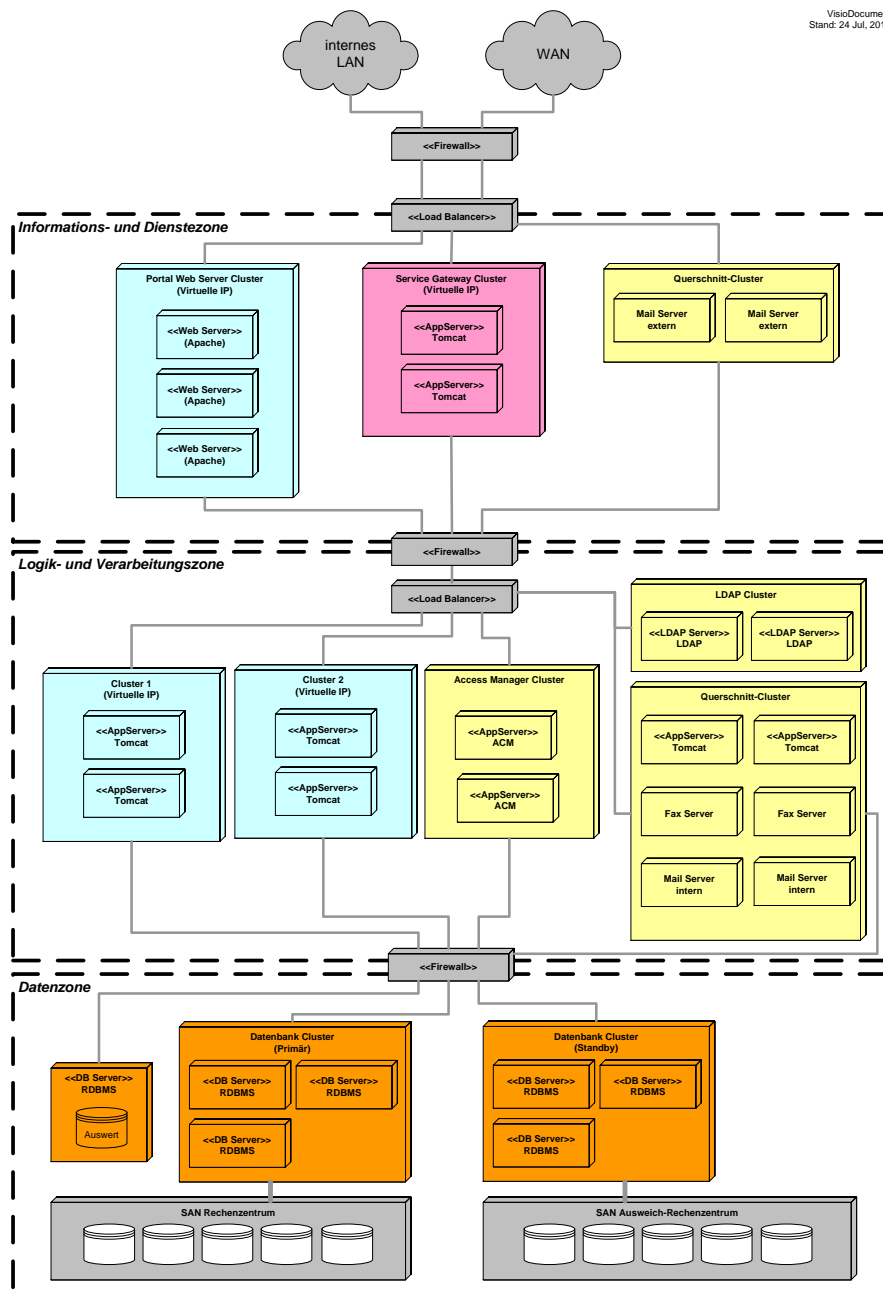


Abbildung 9: TI-Architektur der Produktionsumgebung

Um die Sicherheit in der Datenkommunikation zu gewährleisten, sind die Server unterschiedlichen Sicherheitszonen des Netzwerks zugeordnet. In Abbildung 9 ist eine Sicherheitszone durch ein gestricheltes Rechteck dargestellt. Zonenübergreifende Kommunikationsverbindungen werden von den Firewalls kontrolliert. Damit entspricht die TI-Architektur der Produktionsumgebung den SAGA-Vorgaben.

Der Zugriff durch die Anwender (Clients) und die über externe Systeme angeschlossener Nutzer erfolgt über das WAN bzw. das interne LAN. Die Kommunikation erfolgt per Secure HTTP (HTTPS) mit einem Web-Browser oder ebenfalls per HTTPS oder SMTP via XML- oder Web-Service-Schnittstelle direkt aus der externen Anwendung über ein Service-Gateway-System. Der erste zu passierende Lastverteiler (engl.

Loadbalancer) ist Endpunkt der Verschlüsselung der Kommunikationsverbindung, das heißt innerhalb der Produktionsumgebung kommunizieren die Anwendungssysteme unverschlüsselt miteinander.

Interne Drittsysteme, die aus dem internen LAN mit der IsyFact-Systemlandschaft kommunizieren, tun dies genau wie externe Anwendungen per HTTPS oder SMTP via XML- oder Web-Service-Schnittstellen über ein Service-Gateway-System. Zur Authentifizierung wird ein Access Control Management (ACM) eingesetzt. Dabei greifen alle Applikations- und Web-Server mittels eines installierten ACM-Agenten auf den ACM-Server zu, der in der Logik- und Verarbeitungszone als Querschnittsanwendung betrieben wird.

Administratoren greifen aus dem Admin-Netz direkt mittels Secure-Shell auf die Serversysteme der IsyFact-Systemlandschaft zu (Betriebssystem-Ebene). Aus dem Admin-Netz ist der Zugriff auf die Anwendungen nicht möglich.

Web-Server und Service-Gateway-Systeme kommunizieren mit den Applikationsservern der Logik- und Verarbeitungszone. In Abbildung 9 wird aus Gründen der Vereinfachung davon ausgegangen, dass je Rechnersystem ein Applikationsserver betrieben wird. Zu empfehlen ist allerdings generell die Nutzung eines Rechnersystems mit mehreren Applikationsservern.

Für die Datenhaltung wird ein auf einem relationalen Datenbank-Management-System (RDBMS) basierender Datenbank-Cluster in zweifacher Ausprägung eingesetzt. Eine Ausprägung steuert die Primärdatenbank, die für die operative Bearbeitung von Auskünften und Meldungen zuständig ist. Der operative Datenbestand wird permanent in eine Standby-Datenbank auf dem zweiten Cluster gespiegelt, die für die Datensicherung und für die Erstellung von Auswertungen und Statistiken verwendet wird. Um Auswertungen auf Stichtagsbeständen durchführen zu können, wird ein dedizierter Datenbankserver vorgesehen.

### **5.1.2 Staging-Umgebung (STU)**

Mit dem Begriff „Staging-Umgebung“ werden die Komponenten der technischen Infrastruktur bezeichnet, die zum internen Test verwendet werden und auf denen Probleme des Wirkbetriebs nachgestellt werden können. Eine solche Umgebung ist notwendig, um Problemanalysen durchzuführen und Lösungen für bekannte Probleme vor dem Einsatz im Wirkbetrieb auf ihre Funktionsfähigkeit hin zu prüfen. Die Staging-Umgebung dient auch zu Last- und Performancetests, zur Überprüfung der Installationsroutinen und zur Überprüfung der Ausfallsicherheit. Daher muss sie so ausgelegt sein, dass verlässliche Aussagen im Bezug auf die Produktionsumgebung möglich sind.

Idealerweise ist die Staging-Umgebung eine exakte Kopie der Produktionsumgebung. Häufig ist dies jedoch aufgrund der sehr großen Anzahl Server und den damit verbundenen Investitionskosten für die

benötigte Hardware und Software aus Wirtschaftlichkeitsaspekten nicht sinnvoll.

Daher ist die Staging-Umgebung eine in Bezug auf die Anzahl der Cluster-Knoten kleinere Kopie der Produktionsumgebung. Das heißt, an Stellen, in denen in der Produktionsumgebung ein Cluster mit  $n > 2$  Knoten verwendet wird, wird in der Staging-Umgebung ein Cluster mit 2 Knoten eingesetzt. In der Staging-Umgebung wird auch auf die Datenspiegelung verzichtet. Die Staging-Umgebung einer IsyFact-Systemumgebung ist in Abbildung 10 dargestellt.

Die Server der Staging-Umgebung stehen in eigenen Sicherheitszonen. Die Zonenaufteilung sollte vergleichbar zur Produktionsumgebung sein, da sonst Engpässe in der Netzwerkkommunikation (Bandbreite, Komponentendurchsatz) bei Tests nicht erkannt werden können.

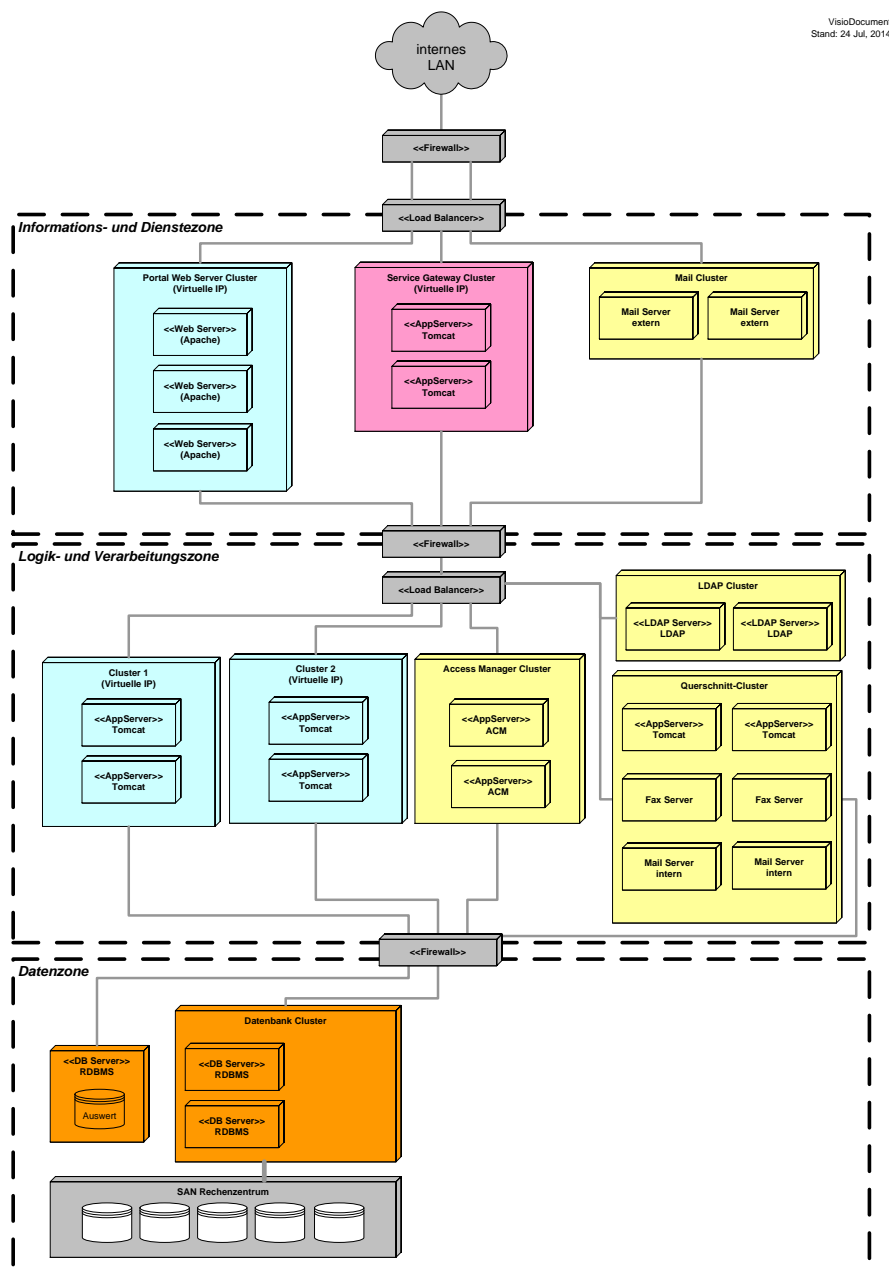


Abbildung 10: TI-Architektur Staging-Umgebung

### 5.1.3 Externe Schulungsumgebung (XSU)

Wie der Name schon sagt wird die externe Schulungsumgebung für die Durchführung von Schulungen verwendet, wobei auch externe Nutzer auf diese Umgebung zugreifen können. Sie ist eine Kopie der externen Testumgebung. Aus diesem Grund wird hier auf Abschnitt 5.1.4 verwiesen.

### 5.1.4 Externe Testumgebung (XTU)

Die externe Testumgebung wird für Tests externer Nutzer verwendet. Damit ist diese Umgebung neben der Produktionsumgebung und der externen Schulungsumgebung die einzige von außen zugängliche Systemumgebung. Abbildung 11 gibt einen Überblick.

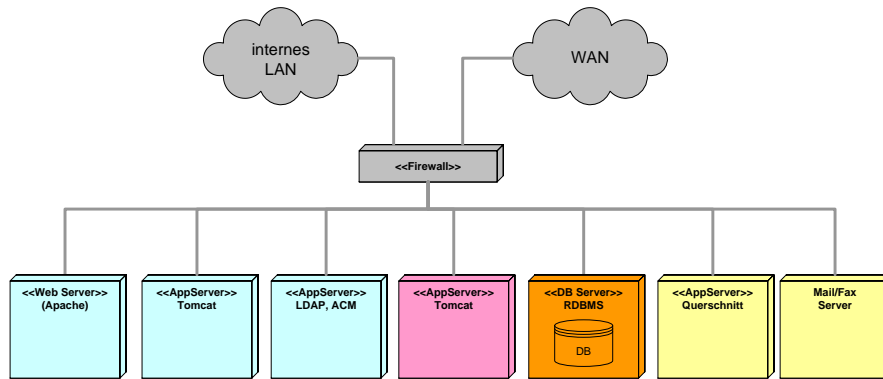


Abbildung 11: TI-Architektur externe Testumgebung

Im Vergleich zur Produktionsumgebung ist die Leistungsfähigkeit dieser Umgebung bei vollständiger Funktionalität deutlich reduziert. Da auch an die Verfügbarkeit der Umgebung geringere Anforderungen gestellt werden, wird auf die Aufteilung in verschiedene Netzwerkzonen und auf den Betrieb der Rechnersysteme im Cluster aus wirtschaftlichen Gründen verzichtet. Die Anwendungssysteme laufen dann auf einzelnen Rechnerknoten ab.

#### 5.1.5 Abnahmetestumgebung (ATU)

Die Abnahmetestumgebung wird zur Durchführung von funktionalen, das heißt fachlichen Abnahmetests genutzt. Sie ist eine Kopie der Entwicklungstestumgebung. Aus diesem Grund wird hier auf Abschnitt 5.1.6 verwiesen.

#### 5.1.6 Entwicklungstestumgebung (ETU)

Die Entwicklungstestumgebung (ETU) wird zur Durchführung von technischen Tests genutzt. An diese Umgebung sind keine Anforderungen an hohe Ausfallsicherheit und Leistungsfähigkeit gestellt. Die Leistungsfähigkeit kann sogar noch unter der der Schulungs- und externen Testumgebung liegen, da davon auszugehen ist, dass die Tests nur von sehr wenigen gleichzeitig aktiven Benutzern durchgeführt werden.

Die Rechnersysteme der Entwicklungstestumgebung werden nur vom internen LAN aus genutzt. Es gibt keine weitere Unterteilung in Sicherheitszonen. Abbildung 12 gibt einen Überblick.



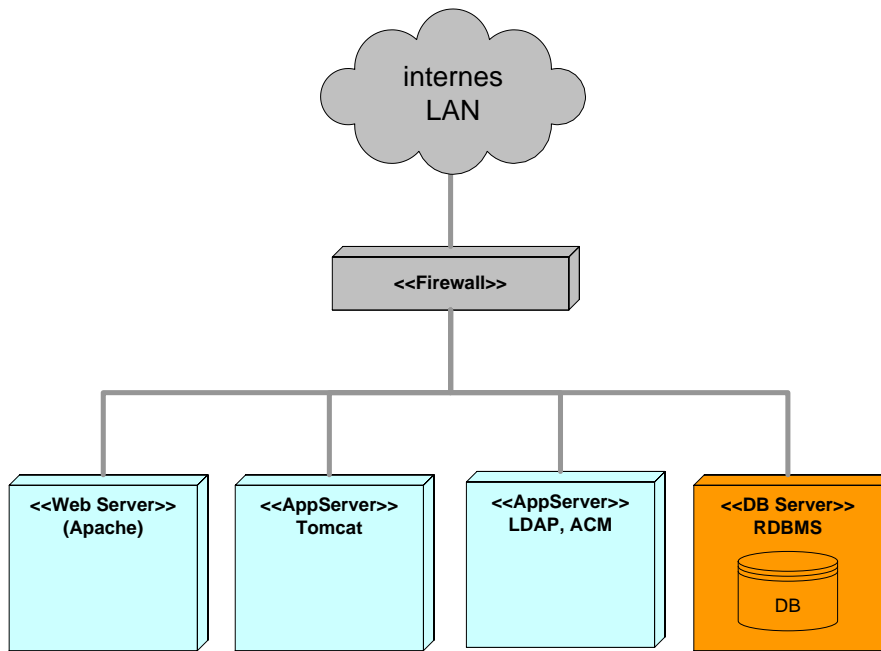


Abbildung 12: TI-Architektur Entwicklungstestumgebung

## 5.2. Minimalanforderungen an die Ablaufumgebung

Als Ablaufumgebung benötigen die gemäß der Referenzarchitektur (siehe Kapitel 4) erstellten IT-Systeme einen Tomcat Servlet-Container. Die Vorgaben zur Konfiguration des Tomcat sind im Dokument [ApacheTomcatKonzept] beschrieben.

Service-Gateway-Systeme und das Portal benötigen zusätzlich noch einen Apache-Webserver.

## 6. Betriebliche Aspekte

In diesem Abschnitt wird auf wichtige Aspekte des Betriebs einer Systemlandschaft eingegangen. Dazu gehören Verfügbarkeit, Ausfallsicherheit, Performance, Lastverteilung, Auslieferung, Installation, Versionswechsel, Monitoring und Datensicherung.

### 6.1. Verfügbarkeit und Ausfallsicherheit

Die Verfügbarkeit wird über einen Prozentwert gemessen, der unter Berücksichtigung von definierten Betriebs- und Wartungsfenstern angibt, wie ausfallsicher ein System sein soll und tatsächlich ist. Abbildung 13 gibt einen Überblick, was bei der Messung der Verfügbarkeit zu berücksichtigen ist.

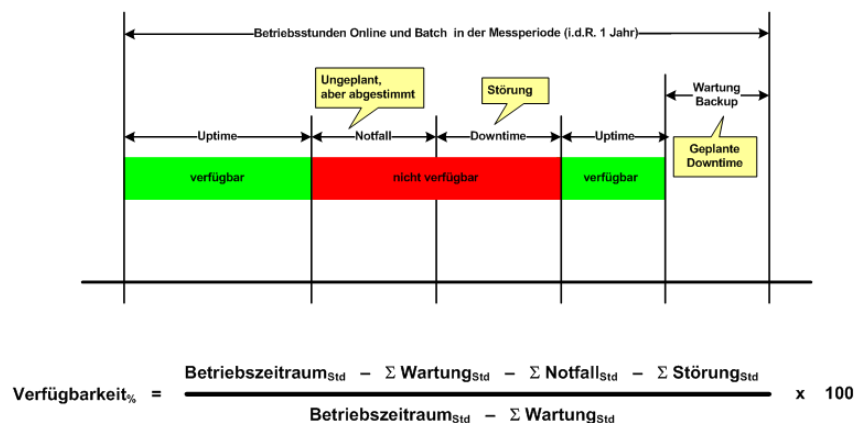


Abbildung 13: Verfügbarkeit

Der Aufbau der Produktionsumgebung ist so gestaltet, dass eine Verfügbarkeit von 7x24 Stunden und eine auf den Monat berechnete Verfügbarkeit von über 99% erreicht werden kann.

In der Produktionsumgebung einer IsyFact-Systemlandschaft wird die geforderte Verfügbarkeit über das Mittel der Redundanz realisiert. Jede Komponente der technischen Infrastruktur ist mindestens zweimal vorhanden. Auf einem Rechnersystem können mehrere Instanzen eines Servers (Applikationsserver, Datenbankserver) betrieben werden. Bei einem Ausfall einer Komponente kann der Wirkbetrieb immer noch über die andere Komponente abgewickelt werden (Failover). Die Zustandslosigkeit der Applikationsserver (siehe Kapitel 4) unterstützt diese Redundanz und das Failover: Beim Ausfall eines Applikationsservers sind die notwendigen Informationen, um den Betrieb über einen anderen Applikationsserver abzuwickeln, beim Ausfall des Knotens nicht verloren gegangen.

Die Server-Knoten der technischen Infrastruktur sind über verschiedene Standorte, Racks, Enclosures verteilt. Dadurch ist die Verfügbarkeit auch in Katastrophen-Szenarien (zum Beispiel dem Brand an einem der beiden Standorte) sichergestellt.

## 6.2. Performance

Ein weiterer wichtiger Aspekt der technischen Infrastruktur ist die Performance der IT-Systeme in einer IsyFact-Systemlandschaft. Auf der Ebene der technischen Infrastruktur wird die Performance vor allem durch die folgenden Aspekte limitiert:

**Leistungsfähigkeit eines Serverknotens:** Die Leistungsfähigkeit eines Serverknotens wird bestimmt durch seine Rechenleistung (Anzahl und Taktung der Prozessoren) und Größe des Hauptspeichers.

**Netzwerk-Durchsatz:** In einem verteilten System erfolgt die Verarbeitung von Informationen innerhalb einer Geschäftsfunktion meist durch die Zusammenarbeit von verschiedenen Knoten der Infrastruktur. Dazu müssen die Knoten miteinander kommunizieren. Ist die Obergrenze des Netzwerk-Durchsatzes erreicht, so führt dies zu einem Performance-Verlust.

Bringt ein IT-System einen Serverknoten an seine Leistungsgrenzen, so existieren grundsätzlich zwei Möglichkeiten der Skalierung: Horizontale Skalierung und Vertikale Skalierung.

Bei der vertikalen Skalierung wird die Hardware eines Serverknotens durch leistungsfähigere Hardware oder durch einen leistungsfähigeren Serverknoten ersetzt. Bei der horizontalen Skalierung werden zusätzliche Serverknoten eingesetzt, um die Last besser verteilen zu können. Vertikale Skalierung tritt im Laufe der Zeit von selbst auf – da die Entwicklung von leistungsfähigerer Hardware in kurzen Zyklen abläuft. Bei der Beschaffung eines Servers wurde das ursprüngliche Modell vom Hersteller häufig durch ein leistungsfähigeres ersetzt. Im Rahmen des Prozesses der gezielten Erneuerung der Hardware (zum Beispiel alle fünf Jahre) bietet die Veränderung der Leistungsparameter dann in der Regel auch Chancen zur Konsolidierung (weniger Server), vorausgesetzt der Ressourcenbedarf der Anwendungen wächst nicht durch neue oder geänderte Anforderungen.

Voraussetzung für die Möglichkeit der horizontalen Skalierung ist eine Software-Architektur, bei der es keine Rolle spielt, welcher Serverknoten einen Verarbeitungsschritt durchführt. Die Referenzarchitektur unterstützt durch den zustandslosen Anwendungsserver den Ansatz der horizontalen Skalierung optimal. Sie bietet für die Zukunft maximale Flexibilität.

## 6.3. Lastverteilung

Für die horizontale Skalierung und die Ausfallsicherheit wird eine Lastverteilung notwendig. Lastverteilung (engl. Loadbalancing) kann entweder durch dedizierte Hardware-Komponenten (Hardware-Lastverteilung) oder durch eine Software-Lösung (Software-Lastverteilung) implementiert werden.

In der Hardware-Lastverteilung werden die eingehenden Anfragen von einer Hardware-Komponente entgegengenommen. Diese Hardware-Komponente ist für die Verteilung der Anfragen auf die dahinterliegenden

Server zuständig. Gleichzeitig kann die Hardware-Komponente die Entschlüsselung der SSL-verschlüsselten Kommunikation übernehmen und entlastet so die Server von dieser Aufgabe.

Analog arbeitet die Software-Lastverteilung. Hier übernimmt eine Software-Komponente wie z. B. ein Http-Server mit entsprechendem Plugin die Verteilung der Anfragen.

#### **6.4. Auslieferung, Installation und Versionswechsel**

Durch die Auslieferung und Installation einer neuen Version eines IT-Systems wird die Konfiguration der Systemlandschaft verändert. In diesem Abschnitt wird auf diese Aspekte eingegangen. Dazu werden zunächst die Begriffe Konfiguration und Auslieferung definiert. Im Anschluss daran werden die organisatorischen Verantwortlichkeiten betrachtet.

**Konfiguration:** Mit dem Begriff „Konfiguration“ wird die Betriebsumgebung zu einem bestimmten Zeitpunkt beschrieben. Alle Komponenten der Betriebsumgebung, das heißt Hardware-Komponenten, Systemsoftware-Komponenten, Anwendungssoftware-Komponenten und ihre Konfigurationsparameter haben eine Version. Eine Konfiguration beschreibt die Betriebsumgebung durch die Angabe der Versionen der einzelnen Komponenten zu einem Zeitpunkt. Wird an der Betriebsumgebung eine Änderung zum Beispiel durch Austausch einer Hardware-Komponente durchgeführt, dann erhält diese Hardware-Komponente eine neue Versionsnummer. Gleichzeitig hat sich die Konfiguration der Betriebsumgebung verändert und wird ebenfalls mit einer neuen Versionsnummer bezeichnet. Gleiches gilt, wenn sich zum Beispiel die Parameter eines Anwendungsservers ändern: Eine neue Version der Parameter liegt vor und damit liegt auch eine neue Konfiguration der Betriebsumgebung vor. Um nachvollziehen zu können, was sich wann warum geändert hat, ist es empfehlenswert, dass jede Veränderung an der Betriebsumgebung genehmigt und nachvollziehbar dokumentiert wird.

**Auslieferung:** Mit dem Begriff „Auslieferung“ wird die Übergabe von Liefergegenständen aus der Hoheit der Software-Entwicklung in die Hoheit des Betriebs bezeichnet. Dabei werden im allgemeinen Software, Parameter und eine Dokumentation ausgeliefert. Bei der Software wird eine ablauffähige Einheit (RPM-Paket) ausgeliefert. Dieses RPM-Paket wird durch das Installationsprogramm unter Angabe der Parameter installiert. Anschließend wird die installierte Software gemäß der in der Auslieferungsdokumentation angegebenen Parameter konfiguriert. Die Auslieferungsdokumentation besteht aus einem Releaseletter, einem Betriebshandbuch und ggf. einem Benutzerhandbuch. Im Releaseletter werden Inhalt und Version der Pakete in Form einer Stückliste beschrieben. Weiterhin wird beschrieben, welche bekannten Fehler und Probleme es mit diesem Paket gibt. Im Releaseletter sind auch die Installationsanleitung inklusive der Parameter der Installation und die Aufstellung der mit dem Release geschlossenen Fehler-Meldungen enthalten. Eine

Auslieferung umfasst ebenfalls Ergänzungen und Anpassungen der betroffenen Betriebshandbücher. Details zum Deployment sind im Dokument [DeploymentKonzept] beschrieben.

Wie oben bereits erwähnt bezeichnet der Begriff Auslieferung einen Verantwortungsübergang zwischen zwei Organisationseinheiten. Die Entwicklung erstellt und testet die Software-Pakete und erzeugt die zugehörigen Dokumentationen. Der Betrieb übernimmt diese Artefakte. Für die Installation der Software ist der Betrieb verantwortlich. Insbesondere liegt die Sicherstellung der Rücksetzbarkeit nach einer fehlgeschlagenen Installation eines Pakets in der Verantwortung des Betriebs. Treten Fehler auf, so informiert der Betrieb die Entwicklung über ein geregeltes Verfahren über das Problem. Auf Anfrage des Betriebs unterstützt die Entwicklung direkt bei der Fehleranalyse. Der Betrieb ist auch für die Dokumentation der Konfigurationsänderungen durch die Installation der neuen Anwendungssoftware-Pakete zuständig.

## **6.5. Betriebsüberwachung**

Um den Gesamt-Status eines Systems zu überwachen, sind die Überwachung der IT-Systeme sowie das Monitoring der Komponenten der Technischen Infrastruktur notwendig. Auf die Überwachung der IT-Systeme wird im Dokument [ÜberwachungKonfigKonzept] im Detail eingegangen. Hier wird unter anderem festgelegt, welche Informationen ein nach der Referenzarchitektur erstelltes IT-System für die Überwachung mindestens bereitstellen muss. Die Schnittstellentechnologie, über die die Informationen der einzelnen Anwendungen zur Überwachung angeboten werden, ist dabei der Standard JMX, der in alle gängigen Überwachungstools integrierbar ist.

Als zentrales Überwachungstool kann z. B. das OpenSource-Werkzeug Nagios verwendet werden.

Zum Monitoring der Technischen Infrastruktur können für das zentrale Überwachungstool verfügbare Client-Programme verwendet werden. Diese Programme ermitteln den Status und die Leistungsparameter (zum Beispiel CPU-Auslastung, Hauptspeicher-Auslastung, Netzwerk-Auslastung) eines Server-Knotens und senden diese Informationen an das zentrale Überwachungstool. Zusätzlich ist auch ein Betriebsmonitoring der Netzwerkkomponenten (Router, Switches und andere) notwendig.

Beim Über- oder Unterschreiten bestimmter Schwellwerte oder bei einem Totalausfall von Komponenten kann vom zentralen Überwachungstool ein Alarm ausgelöst und der zuständige Systembetreuer informiert werden. Es müssen Abhängigkeitsgraphen für die Rechnersysteme und Anwendungen erstellt werden, an Hand derer das zentrale Überwachungstool sinnvolle Alarmierungen vornehmen kann. Auch müssen darüber hinaus – am besten bereits während der Entwicklung – sicherheitskritische Ereignisse definiert werden, deren Auftreten explizit überwacht werden soll.

Aus Sicherheitsgründen kommunizieren die Client-Programme des zentralen Überwachungstools nicht direkt mit dem zentralen Überwachungstool, sondern mit einem Satelliten-System. Damit benötigt nicht jeder Server der Betriebsumgebung eine direkte Verbindung zum plattformübergreifenden zentralen Überwachungstool.

## **6.6. Daten und Datensicherungen**

Die Datensicherung (Backup) für eine IsyFact-Systemlandschaft erfolgt durch den Betrieb. Bei der Datensicherung muss zwischen der Sicherung der Software inklusive der Konfigurationsdateien, der Sicherung der Logdateien und der Sicherung der Datenbank unterschieden werden.

### **6.6.1 Software und Konfigurationsdateien**

Mit Ausnahme des Betriebssystems wird die Software selbst inklusive zugehöriger Konfigurationen über ein Konfigurations-Management verwaltet. Dieses hat nichts mit den Systemen zum Betrieb der Anwendungen zu tun und wird unabhängig betrieben und gesichert. Jeder Stand ist über das Konfigurations-Management vollständig reproduzierbar. Im Falle von Datenverlust können die Software-Pakete in den entsprechenden Versionen neu gebaut und neu installiert werden. Dies wird aber in der Regel nur dann notwendig, wenn auch die Systemsicherungen vom Verlust betroffen sind. Der Betrieb muss jedoch in der Lage sein, bei Ausfall eines Server-Knotens eine Neuinstallation in kurzer Zeit durchführen zu können (zum Beispiel durch Einspielen zuvor gesicherter Images).

### **6.6.2 Log-Dateien**

Die verschiedenen Server der Betriebsumgebung schreiben Log-Dateien in das Dateisystem auf dem jeweiligen Server. Die Log-Dateien enthalten wichtige Informationen, um bei Problemen das Verhalten der Anwendung nachzuvollziehen oder Nachweise zu erbringen. Sie müssen daher gesichert werden. Gemäß einer Anforderung des Bundesamtes für Sicherheit in der Informationstechnik (BSI), die im behördlichen Umfeld verbindlich ist, müssen die Log-Dateien zentral gesichert werden.

Die Log-Dateien eines Servers werden regelmäßig von einem Scheduler-Job auf einen zentralen Log-Server transferiert, von wo aus sie gesichert werden. Der Betrieb ist für die Datensicherung der Log-Dateien verantwortlich. Durch die Vorgaben zum Logging wird sichergestellt, dass Log-Dateien zusammengeführt werden können, und dass der technische Ablauf auch über verschiedene Log-Dateien hinweg nachvollzogen werden kann. Die Grundidee dabei ist, dass alle Log-Dateien in einem einheitlichen Format vorliegen und zusätzlich jeder Anfrage an das Gesamtsystem eine spezielle ID (correlation id) zugeordnet wird, mit der sich zusammen gehörende Einträge unterschiedlicher Log-Files einander zuordnen lassen.

### **6.6.3 Datenbank**

Die Geschäftsdaten aller Anwendungen einer IsyFact-Systemumgebung werden in relationalen Datenbanken gehalten. Der Verlust dieser Daten

wäre mit erheblichen Schaden verbunden. Eine angemessene Datensicherung ist daher unerlässlich.

Da nur die Produktionsumgebung Echtdaten verarbeitet, beschränken wir uns nachfolgend auf die Datensicherung in der Produktionsumgebung. Die Sicherung der Daten in den anderen Systemumgebungen muss nicht regelmäßig erfolgen, sondern kann punktuell je nach Bedarf durchgeführt werden. Der durchgängige Einsatz eines Datenbanksystems vereinfacht dabei die Datensicherung. Neben den in der Datenbank gespeicherten Daten müssen auch andere Datenspeicher, wie z.B. die LDAP-Daten oder Quellen bei der Datensicherung berücksichtigt werden.

Alle Geschäftsdaten werden redundant in Primär- und Standby-Datenbanken gehalten<sup>2</sup>. Zu jeder Primärdatenbank wird mindestens eine Standby-Datenbank eingerichtet. Die Primärdaten werden in einem SAN in einem Rechenzentrum abgelegt, die Standby-Daten im besten Fall in einem SAN eines anderen Rechenzentrums. Fällt die Primärdatenbank aus, kann ihre Aufgabe von der Standby-Datenbank übernommen werden (Failover). Die Verfügbarkeit der Anwendungen wird dadurch deutlich erhöht, denn das Failover dauert nur Sekunden oder Minuten, während das Zurückspielen einer Datensicherung (Restore) in der Regel deutlich länger dauern wird. Ein Restore wird nur dann notwendig sein, wenn die Datenbestände aufgrund von Softwarefehlern oder menschlichem Versagen korumpiert wurden. Ein Restore aufgrund von technischen Gründen ist sehr unwahrscheinlich. Auch ein Komplettausfall der Primärdatenbank ist durch den Einsatz eines über Rechenzentren verteilten Datenbank-Clusters sehr unwahrscheinlich.

Die Replikation der Daten aus den Primärdatenbanken in die Standby-Datenbanken erfolgt über die Weitergabe der ReDo-Informationen. Die Datensicherung auf Bänder erfolgt online ausschließlich auf den Standby-Datenbanken und belastet die Primärdatenbank nicht. Die Datensicherung der Produktionsdaten wird vom Betrieb durchgeführt und verantwortet.

---

<sup>2</sup> Für besonders ausfallsichere Systeme wird ein Primär- und Sekundär-Datenbank-Cluster verwendet, z. B. ein Oracle RAC.

## 7. Vereinfachte Varianten der Referenzarchitektur

In diesem Kapitel wird vorgestellt, wie sich die Referenzarchitektur der IsyFact in Anwendungsszenarien mit reduzierten Anforderungen nutzen lässt. Dazu wird zunächst motiviert, welche Szenarien das sind und warum eine Nutzung der IsyFact-Referenzarchitektur dort möglich ist. Anschließend werden die Vereinfachungen – der Verzicht auf einen SOA-Ansatz und eine vereinfachte TI-Architektur – vorgestellt. Danach wird kurz vorgestellt, wie eine vereinfachte Architekturform zu einer kompletten Plattform ausgebaut werden kann. Abschließend werden die Vereinfachungen am Beispiel von Kleinverfahren dargestellt.

### 7.1. Einführung

In der bisher vorgestellten Form ist die Referenzarchitektur für die Umsetzung großer Systemlandschaften ausgelegt, also für den Einsatz in großen Anwendungen mit hohem Funktionsumfang und hohen Anforderungen an Verfügbarkeit, Performance und Lastverhalten. Deshalb folgt sie einem serviceorientierten Plattformansatz. Bei geringeren Anforderungen an die Ausbaufähigkeit und geringeren nichtfunktionalen Anforderungen können hier Vereinfachungen vorgenommen werden, die im Folgenden dargestellt werden: der Verzicht auf eine SOA-Plattform und eine vereinfachte technische Infrastruktur.

### 7.2. Vereinfachung durch Verzicht auf eine SOA-Plattform

Die vorgestellte Plattform-Architektur der IsyFact bietet verschiedene Vorteile:

- Bereitstellen zentraler Services, z.B. der Druck auf einer Druckstraße oder alphanumerische Suche. Diese Services können einfach aus verschiedenen Anwendungen heraus genutzt werden. Sie werden oft durch Produkte umgesetzt und werden in der Regel als separate Prozesse betrieben. Hier vereinfacht die Referenzarchitektur die Nutzung dieser Services.
- Nutzung zentraler Datenbestände, z. B. durch ein zentrales Schlüsselverzeichnis.
- Bessere Handhabbarkeit der einzelnen Anwendungen: Die in den Anwendungen umgesetzte Fachlichkeit ist in der Regel sehr umfangreich. Daher bietet die Auflösung in Services ein wichtiges Mittel, um die Komplexität der einzelnen Systeme zu reduzieren und sie somit langfristig wartbar und weiterentwickelbar zu halten.
- Flexibilität und Erweiterungsfähigkeit: Services einer Fachanwendung können durch die serviceorientierte Architektur leicht in andere Fachanwendungen eingebunden werden.

In einigen Fällen bestehen diese Anforderungen nicht, daher wäre dann der Aufbau und Betrieb einer SOA-Plattform nicht angemessen. In diesem Fall können Anwendungen auch in einer kompakten Form gebaut werden.



Dies ist sinnvoll, wenn der Funktionsumfang der Anwendung beschränkt ist, keine zentralen Services oder Datenbestände genutzt werden sollen und keine größeren Erweiterungen in der Zukunft zu erwarten sind. Dieses Szenario ist in Abbildung 14 dargestellt.

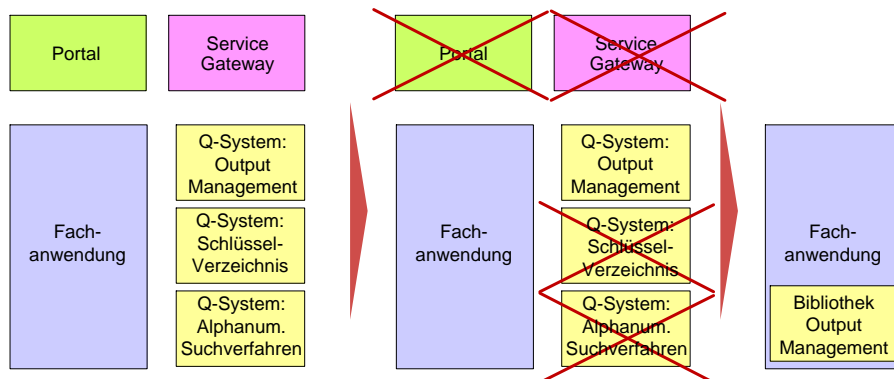


Abbildung 14: Vereinfachung der SOA-Architektur

Links ist eine typische Umsetzung einer Fachanwendung in der IsyFact-Systemlandschaft zu sehen. Sie ist über ein Portal in eine Unternehmenslandschaft integriert, hat Außenschnittstellen zu anderen Organisationen, die über ein Service Gateway angeboten werden und sie nutzen die zentralen Services „Output Management“, „Alphanumerisches Suchverfahren“ und das zentrale Schlüsselverzeichnis.

In der Mitte ist dargestellt, dass es Anwendungen gibt, die diese Anforderungen nicht haben:

- Sie werden stand-alone genutzt und sind nicht in ein Portal integriert.
- Sie haben keine externen Schnittstellen.
- Sie greifen nicht auf zentrale Schlüsseldaten zu.
- Sie benötigen kein Alphanumerisches Suchverfahren.
- Das Output Management wird lediglich dazu benötigt, PDF-Dokumente zu erzeugen. Ein Ausdruck über eine zentrale Druckstraße erfolgt nicht.

In einem solchen Fall kann eine kompakte Anwendung erstellt werden, die das Querschnittssystem „Output Management“ nicht via Service-Aufruf anspricht, sondern dieses als Bibliothek einbindet. Die Software-Referenzarchitektur der Anwendung (vgl. Kapitel 4) bleibt trotzdem unverändert.

### 7.3. Vereinfachungen in der technischen Infrastruktur

Die technische Infrastruktur kann an zwei Stellen vereinfacht werden. Zum einen ist es möglich, auf den Clusterbetrieb zu verzichten, zum anderen

kann die Aufteilung in SAGA-konforme Zonen weggelassen werden. Dies wird im Folgenden ausgeführt

### 7.3.1 Installation ohne Clustering

Bei geringeren nicht-funktionalen Anforderungen kann auch nur eine Anwendungsinstanz zum Einsatz kommen und kein Cluster für die Datenbank verwendet werden. Dies ist in Abbildung 15 dargestellt.

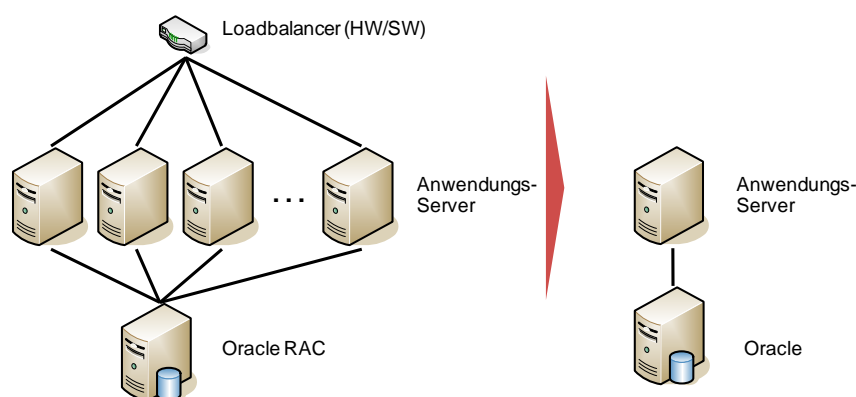


Abbildung 15: Vereinfachung durch Wegfall von Clustering

### 7.3.2 Verzicht auf das SAGA-Zonenmodell

Weiterhin sieht die Referenzarchitektur der Technischen Infrastruktur ein Zonenmodell gemäß SAGA vor. Dieses Zonenmodell unterstützt eine klare Sicherheitsarchitektur und ist in Kapitel 5 beschrieben. Durch die Aufteilung in Informations- & Dienstzone, Logik- & Verarbeitungszone sowie Datenzone sind die Grundlagen geschaffen, um z.B. E-Government-Anwendungen sicher betreiben zu können.

Der SAGA-Standard in der Version 4 gilt grundsätzlich für E-Government-Anwendungen. In anderen Bereichen wird der SAGA-Standard nur dann empfohlen, wenn die Kosten/Nutzen-Betrachtung positiv ausfällt. In den Situationen, wo dies nicht der Fall ist, steht damit der Architekt eines Systems vor der Entscheidung, eine andere Architektur für die technische Infrastruktur zu wählen. Die Referenzarchitekturen der IsyFact ermöglichen auch eine solche, nicht SAGA-konforme Umsetzung. Sinnvoll ist dies allerdings nur, wenn die Anwendung mit einer vereinfachten Software-Architektur konzipiert ist, d. h. wenn sie nicht als SOA-Plattform konzipiert ist.

Mögliche, alternative Architekturen für die technische Infrastruktur sind:

- **Stand-Alone-Anwendung auf einem Nutzerrechner:** Hierbei wird die Anwendung wie jedes andere Programm auch lokal auf einem Rechner installiert. Dies ist möglich, wenn die Anwendung über eine native GUI verfügt.
- **Zentral auf einem nur intern zugreifbaren Server:** Hier wird die Anwendung auf einem zentralen Server bereitgestellt, der z. B. nur für die Nutzer einer bestimmten Abteilung zugreifbar ist. Diese

Installation ist möglich, wenn die Anwendung über eine Web-GUI verfügt.

#### **7.4. Erweiterungsmöglichkeiten einer vereinfachten Architektur**

Wenn eine Anwendung nach einer vereinfachten Architektur entwickelt wurde, ist die Erweiterung zu einer SAGA-konformen, serviceorientierten Anwendung möglich, wenn vorher gewisse Randbedingungen beachtet wurden.

##### **7.4.1 Installation innerhalb einer SAGA-konformen Plattform**

Um eine Anwendung in einer SAGA-konformen Plattform zu betreiben, muss unterschieden werden, ob es sich um eine native Java-GUI handelt oder um eine Web-GUI. Web-GUIs werden ja schon auf einem zentralen Server betrieben und können ohne weiteres in einer SAGA-konformen 3-Zonen-Umgebung betrieben werden.

Im Falle einer Nativen GUI ist der Umbau aufwändiger. Die Anwendung muss die Anwendung in zwei Teile zerlegt werden. Ein Teil enthält die eigentliche GUI, der andere Teil den Rest der Anwendung, d. h. Anwendungskern und Datenzugriff. Die beiden Teile müssen dann über interne Service-Aufrufe miteinander kommunizieren. Das bedeutet, dass alle Methoden-Aufrufe zwischen den Schichten Anwendungskern und GUI per HTTP-Invoker erfolgen.

##### **7.4.2 Installation innerhalb eines Clusters**

Um eine Anwendung innerhalb eines Clusters von Anwendungsservern zu betreiben, ist es notwendig, den Bearbeitungszustand nicht im Hauptspeicher des Serverprozesses zu halten, sondern ihn bei jedem Request in der Datenbank zu persistieren. Dies ist in den GUI-Referenzarchitekturen der IsyFact vorgegeben, allerdings werden Verletzungen dieses Prinzips nicht auffallen, wenn man eine vereinfachte Anwendung auf nur einem Serverknoten betreibt. Daher ist vor dem Umzug auf einen Cluster die Anwendung durch geeignete Tests oder Code-Inspektionen zu überprüfen, ob konform zu den Vorgaben entwickelt wurde. Alternativ kann natürlich auch auf die Persistierung des Anwendungszustands verzichtet werden, wenn man bewusst auf die Installation in einem Cluster verzichten will.

##### **7.4.3 Installation innerhalb einer SOA**

Der Nutzung einer Anwendung innerhalb einer SOA ist problemlos möglich. Falls Services der Plattform genutzt werden sollen, können Aufrufe dieser Services ohne weitere Vorkehrungen implementiert werden. Genauso ist es möglich, innerhalb der Plattform Services anzubieten.

#### **7.5. Anwendungsbeispiel: Kleinverfahren**

Unter einem Kleinverfahren versteht man im öffentlichen Sektor Anwendungen, die nur geringe Anforderungen an nichtfunktionale Eigenschaften haben. Typisch für solche Anwendungen sind:

- Sie werden in der Praxis oft durch Implementierungen auf Excel- oder Access-Basis umgesetzt.
- Sie haben nur wenige Nutzer
- Sie werden nur intern innerhalb einer Organisation genutzt.
- Sie haben nur geringe Anforderungen an die Sicherheitsarchitektur.
- Sie stehen von ihrer Funktionalität her für sich selbst und sind nicht Bestandteil einer Plattform.

Ein solches Kleinverfahren kann mit einer vereinfachten Architektur umgesetzt werden. Dies umfasst:

- Verzicht auf die SOA-Plattform (Abschnitt 7.2)
- Verzicht auf Clustering (Abschnitt 7.3.1)
- Verzicht auf das SAGA-Zonenmodell (Abschnitt 7.3.2)

Diese Vereinfachungen sind für ein Kleinverfahren nicht zwingend, sondern sie dienen dazu, das Verfahren in einem angemessenen Kosten-Nutzen-Verhältnis zu erstellen. Falls die Kosten-Nutzen-Betrachtung es erlaubt, kann auf die Vereinfachungen natürlich auch verzichtet werden.

## 8. Quellenverzeichnis

### [AnwendungskernDetailkonzept]

Detailkonzept der Komponente Anwendungskern  
10\_Blaupausen\technische\_Architektur\Detailkonzept\_Komponente\_Anwendungskern.pdf.

### [ApacheTomcatKonzept]

Konzept Apache Tomcat (INTERN)

### [BatchDetailkonzept]

Detailkonzept Komponente Batch  
10\_Blaupausen\technische\_Architektur\Detailkonzept\_Komponente\_Batch.pdf.

### [Berechtigungskonzept]

Berechtigungskonzept  
Muss projektspezifisch erstellt werden.

### [DatenzugriffDetailkonzept]

Detailkonzept Komponente Datenzugriff  
10\_Blaupausen\technische\_Architektur\Detailkonzept\_Komponente\_Datenzugriff.pdf.

### [DeploymentKonzept]

Konzept Deployment für IsyFact-Anwendungen  
30\_Plattform\Konzept\_Deployment.pdf.

### [IsyFactEinstieg]

IsyFact-Einstieg  
00\_Allgemein\IsyFact-Einstieg.pdf.

### [IsyFactReferenzarchitekturITSystem]

IsyFact – Referenzarchitektur für IT-Systeme  
00\_Allgemein\IsyFact-Referenzarchitektur-IT-System.pdf.

### [IsyFactSystementwurf]

Vorlage für Systementwürfe  
40\_Methodik\20\_Systementwurf\IsyFact-Vorlage\_Systementwurf.dot.

### [ProduktKatalog]

IsyFact Produktkatalog  
00\_Allgemein\Produktkatalog.pdf.

### [SAGA40]

SAGA Version 4.0 – Standards und Architekturen für E-Government-Anwendungen;  
Publikation der KBSt; März 2008.  
<http://www.kbst.bund.de/saga>.

### [ServiceDetailkonzept]

Detailkonzept Komponente Service  
10\_Blaupausen\technische\_Architektur\Detailkonzept\_Komponente\_Service.pdf.

### [Styleguide]

Styleguide  
20\_Bausteine\Styleguide\Styleguide.pdf.

[TailoringVorlage]

Vorlage für IsyFact Tailoring Dokument  
00\_Allgemein\IsyFact-Tailoring.dotx.

[ÜberwachungKonfigKonzept]

Konzept Überwachung und Konfiguration  
20\_Bausteine\Ueberwachung\_Konfiguration\Konzept\_Ueberwachung-Konfiguration.pdf.

[WebGUIDetailkonzept]

Detailkonzept Komponente Web-GUI  
10\_Blaupausen/technische\_Architektur/Detailkonzept\_Komponente\_Web\_GUI.pdf.

## 9. Abbildungsverzeichnis

Abbildung 1: Fachliche Architektur einer Anwendungslandschaft .....	7
Abbildung 2: Horizontale Skalierung .....	9
Abbildung 3: Referenzarchitektur der technischen Infrastruktur für die Produktionsumgebung .....	10
Abbildung 4: Durch die IsyFact abgedeckte Bereiche .....	13
Abbildung 5: Anpassung der technischen Referenzarchitekturen auf den fachlichen Kontext .....	16
Abbildung 6: Fachliche Architektur einer Anwendungslandschaft .....	17
Abbildung 7: Aufrufbeziehungen für eine Fachanwendung mit GUI und XML-Schnittstelle .....	24
Abbildung 8: Überblick Systemumgebungen .....	26
Abbildung 9: TI-Architektur der Produktionsumgebung .....	28
Abbildung 10: TI-Architektur Staging-Umgebung .....	31
Abbildung 11: TI-Architektur externe Testumgebung .....	32
Abbildung 12: TI-Architektur Entwicklungstestumgebung .....	33
Abbildung 13: Verfügbarkeit .....	34
Abbildung 14: Vereinfachung der SOA-Architektur .....	41
Abbildung 15: Vereinfachung durch Wegfall von Clustering .....	42