

# RAfEG: Referenz-Systemarchitektur und prototypische Umsetzung <sup>\*†</sup>

– Ausschnitt aus dem Abschlussbericht zum Projekt “Referenzarchitektur für  
E-Government” (RAfEG) –

R. Kunis, G. Rünger  
Fakultät für Informatik  
Technische Universität Chemnitz  
09107 Chemnitz

E-Mail: {krap,ruenger}@informatik.tu-chemnitz.de

---

\*Das Projekt RAfEG wurde mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) im Rahmen der Forschungsoffensive „Software Engineering 2006“ unter dem Förderkennzeichen 01 IS C07 C gefördert. Das Projekt setzte sich zusammen aus dem Institut für Wirtschaftsinformatik (IWi) im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI), der NHConsult GmbH und der Professur Praktische Informatik der Technischen Universität Chemnitz. Weitere Informationen sind auf der Projekthomepage <http://www.rafeg.de> zu finden.

†Dieser Bericht ist ein überarbeiteter Ausschnitt des RAfEG-Projektabschlussberichts. Er umfasst die von der Professur Praktische Informatik der Technischen Universität Chemnitz beigetragenen Teile.



## Zusammenfassung

Das Ziel des RAfEG-Projektes bestand in der Entwicklung einer Referenzarchitektur „E-Government“, die die notwendigen Komponenten zur Realisierung informations- und kommunikationstechnischer Systeme (IuK-Systeme) für typische Prozesse in nachgeordneten Behörden der Innenministerien der Bundesländer bereitstellte. Die Architektur RAfEG stellt einen ganzheitlichen Ansatz dar, der viele wesentliche Aspekte, beginnend mit der formalen Beschreibung der fachlichen Zusammenhänge bis hin zur Entwicklung von verteilt agierenden Softwarekomponenten behördlicher Geschäftsprozesse umfasst. Die Architektur liefert unter Berücksichtigung hardwareseitiger Voraussetzungen die Struktur von Softwarekomponenten zur Verwaltungsautomatisierung. Die Architektur RAfEG wurde als räumlich verteiltes komponentenbasiertes Softwaresystem entworfen. Dabei war es notwendig, Konzepte zur effizienten Nutzung von heterogenen Systemen für interaktive Anwendungen im Bereich E-Government zu entwickeln. Die prototypische Umsetzung der Architektur erfolgte für Planfeststellungsverfahren/Plangenehmigungsprozesse am Beispiel des Regierungspräsidiums Leipzig. Das Vorhaben war geprägt von der Entwicklung eines durchgängigen Konzeptes zur optimalen IuK-technischen Unterstützung von Verwaltungsprozessen. Dies führte von der Modellierung der fachlichen Zusammenhänge (Fachkonzept) über die entwicklungsorientierte, methodische Abbildung der zu implementierenden Sachverhalte (Datenverarbeitungskonzept) bis zur komponentenbasierten Softwareentwicklung (Implementierungskonzept). Dieses Konzept mündete in einer Referenzarchitektur für typische E-Government-Prozesse. Dazu wurden neben den rein fachlichen, aufgabenbezogenen Aspekten insbesondere Sicherheitsaspekte sowie technische und organisatorische Schnittstellen ausführlich betrachtet. Der durchgängige Einsatz von Open Source Software führt hierbei zu einer kosteneffizienten, flexiblen Referenzlösung, die durch ihre komponentenbasierte Struktur als weiteren Aspekt sehr gut an spezielle Anforderungen anpassbar ist.

**Keywords:** E-Government, Verteilte Softwareentwicklung, Software-Architektur, Komponentenbasierte Software, Workflow Management System, Vorgangssteuerung, Verteilte Workflows, Hierarchische Workflows

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Analyse der IUK-Systeme und IUK-technische Realisierungsbedingung</b>	<b>1</b>
2.1	Systemkatalog . . . . .	1
2.2	Kriterienkatalog . . . . .	4
2.3	Bewertung der Systeme . . . . .	5
<b>3</b>	<b>Entwicklung der Referenzarchitektur</b>	<b>6</b>
3.1	Softwarearchitektur-Komponente . . . . .	7
3.2	Infrastrukturarchitektur-Komponente/Integration zu RAFEG . . . . .	11
<b>4</b>	<b>Prototypische Umsetzung (Implementierung)</b>	<b>13</b>
4.1	Protokoll mit Schnittstellen . . . . .	13
4.2	Prototypische Realisierung ohne Sicherheitsaspekte . . . . .	16
4.3	Baukastensysteme . . . . .	18
4.4	Sicherheitsaspekte . . . . .	24
4.5	Externe Datenrepräsentation . . . . .	26
4.6	Anpassung/Evaluierung . . . . .	28
<b>5</b>	<b>Wissenstransfer</b>	<b>28</b>
<b>6</b>	<b>Projektbeteiligte</b>	<b>29</b>

# 1 Einleitung

Die Forschungs- und Entwicklungsaufgaben der Professur Praktische Informatik der Technischen Universität Chemnitz bestanden vorwiegend in der Konzeption und prototypischen Realisierung einer Softwarearchitektur, die die Erstellung von Anwendungssoftware für eine breite Klasse von Applikationen des E-Government unterstützt. Die Arbeiten spannten den Bogen von der Analyse und Bewertung gegebener Informations- und Kommunikationssysteme (IuK-Systeme) im Kontext des E-Government (Abschnitt 2) über die Planung einer Softwarearchitektur zur Unterstützung typischer E-Government-Prozesse (Abschnitt 3) bis hin zur prototypischen Implementierung für heterogene verteilte Plattformen mit austauschbaren Komponenten (Abschnitt 4). Die prototypische Implementierung wurde als Open Source Projekt durchgeführt. Die besondere wissenschaftliche Herausforderung war die durchgängige Behandlung von Prozessen des E-Government, die mit der Modellierung des Referenzprozesses durch die anderen Projektpartner startete und nach geeigneten Transformationen durch die Software-Referenzarchitektur entsprechend der Modellierung ausgeführt wird. Besondere Berücksichtigung hat hierbei der Referenzprozess Planfeststellungsverfahren gefunden. Konzipiert und realisiert wurde die Software-Referenzarchitektur jedoch gemäß Projektziel in einer offenen, erweiterbaren Form, die die durchgängige Bearbeitung einer breiten Klasse von E-Government-Prozessen und darüber hinaus auch ähnlich strukturierten Prozessen ermöglicht. Dies wird im Folgenden detailliert erläutert und vorgestellt.

## 2 Analyse der IuK-Systeme und IuK-technische Realisierungsbedingung

Die Herausforderung des RAfEG Verbundprojektes bestand in der Entwicklung einer Referenzsoftwarearchitektur, die die Erstellung, Anpassung und Integration von Anwendungssoftware zur Unterstützung von organisationsübergreifenden, modernen Verwaltungsvorgängen in einer standardisierten Form ermöglicht. Der erste Schritt umfasste die Erstellung eines Software- und Kriterienkatalogs, in dem bestehende Software bewertet und ihre Tauglichkeit für die Software-Referenzarchitektur nach festgelegten Kriterien geprüft wurde. Weiterhin wurde ein Vorgehensmodell für die Auswahl geeigneter Systeme erstellt. Die Analyse von IuK-Systemen erfolgte dabei unter Betrachtung bestehender Netze und Plattformen innerhalb des Regierungspräsidiums Leipzig und den Trägern öffentlicher Belange. Dazu wurden bestehende Hard- und Softwaresysteme sowie existierende Netzarchitekturen untersucht und innerhalb einer installierten Testumgebung hinsichtlich zukünftig nutzbarer IT-Systemumgebungen abgeglichen.

### 2.1 Systemkatalog

Die Zusammenstellung des Systemkatalogs erfolgte hinsichtlich der Systemklassen Bürokommunikationsmanagement-, Dokumentenmanagement- (insb. Dokumentenarchivierung), Workflowmanagement- und Geografische Informationssysteme sowie Sicherheitssysteme – mit einer Konzentration auf Verschlüsselungs- und Signatursysteme – und Content-Publishing-Systeme. Neben der Systemauswahl und -bewertung wurde, vorrangig in den beiden letzten Themenfeldern, tiefer auf ggf. konkurrierende Konzepte eingegangen.

Bestehende Sicherheitsprobleme beim Austausch verschlüsselter Informationen zwischen mobilen Endgeräten und einem Server über HTTPS und WTLS (Wireless Transport Layer Security) wurden untersucht und Lösungsvorschläge erarbeitet. Die Nutzung von digitalen Signaturen auf Basis hierarchischer Zertifikatsstrukturen, welche zum Signaturgesetz (SigG) [Bun01a] und der Signaturverordnung (SigV) [Bun01b] konform sind, wurde vor allem unter praktischen Gesichtspunkten betrachtet. Zusätzlich wurde auf entsprechende Empfehlungen des E-Government-Handbuchs [Bun05] des Bundesamtes für Sicherheit in der Informationstechnik (BSI) eingegangen.

Die Erstellung des Systemkatalogs erfolgte auf Grundlage der tatsächlichen Relevanz für öffentliche Verwaltungen, insbesondere des Regierungspräsidiums Leipzig. Für die geplante Softwarearchitektur wurden wünschenswerte Systeme berücksichtigt. Dazu wurden betrachtet:

- Interne und externe Datennetze: Insbesondere in Hinblick auf Ihre Nutzung für den TCP/IP-Protokollstack und der Erfüllung von Sicherheitsstandards.
- Dokumentenmanagementsysteme: Vorrangig die Integration in bzw. Nutzbarkeit durch ein Workflowmanagementsystem (WfMS) unter Nutzung offener Standards.
- Workflowmanagementsysteme: Hinsichtlich ihrer Flexibilität beim Einsatz innerhalb der, zum Zeitpunkt der Erstellung des Systemkataloges noch nicht bekannten, RAfEG-Systemarchitektur.
- Geografische Informationssysteme: Insbesondere in Bezug auf die Unterstützung offener Dokumentenaustauschformate.
- Sicherheitssoftwaresysteme und –frameworks: Unter Berücksichtigung der transparenten Integration in bestehende heterogene IT-Systemumgebungen.
- Bürokommunikationssoftware: In Hinblick auf den organisationsübergreifenden Dokumentenaustausch bei Unterscheidung der Anwendungsfälle der Dokumentenerstellung und –modifikation von der Dokumentenverwendung und –archivierung.

Besondere Berücksichtigung fanden Open Source Systeme, da diese zumeist über offen Schnittstellen verfügen, die Anpassbarkeit durch die offenen Quellen gegeben ist und der Kostenaspekt in Verwaltungen eine nicht zu unterschätzende Rolle spielt. Im Folgenden werden die Kernpunkte des Systemkatalogs kurz dargestellt.

**Dokumentenmanagementsysteme (DMS):** Das Dokumentenmanagement befasst sich mit der Erfassung/Erstellung, Bearbeitung, Speicherung, Weitergabe und Archivierung von elektronischen Dokumenten. DMS müssen dabei verschiedene Grundanforderungen erfüllen, z.B. die Erstellung oder Erfassung von neuen Dokumenten mit der Möglichkeit Zusatzinformationen für jedes Dokument zu speichern. Änderungen von Dokumenten mit der Möglichkeit der Speicherung von Versionen eines Dokumentes müssen durchführbar sein, um die Änderungen nachvollziehen bzw. rückgängig machen zu können. Im DMS gespeicherte Dokumente müssen zwischen Nutzern verteilt werden können und eine Archivierung aller Versionen eines Dokuments muss möglich sein. Als Schnittstellen zur Integration in eine bestehende Softwareinfrastruktur haben sich die folgenden Standards etabliert, die im Systemkatalog [BHRV05] genauer betrachtet werden: ODMA, TWAIN, OCR, LDAP, WebDAV [GWF<sup>+</sup>99], MAPI, XML [W3C07a] und [Ado04]. Bei den existierenden DMS wurden sowohl freie Open Source Systeme, wie Knowledge Tree [Jam07], Jakarta Slide [Apa] und OpenCMS [Alk07], als auch kommerzielle Systeme, wie e-komm

Office [n-k07], cc DMS [CC 07] und SAPERION [SAP07] betrachtet. Bei den Anforderungen an Dokumentenmanagementsysteme für den Einsatz in öffentlichen Verwaltungen spielen zusätzlich gesetzliche Anforderungen, wie z.B. Revisionsicherheit und die Zertifizierung nach ISO9000, eine wichtige Rolle.

**Bürokommunikationssoftware (BKS):** Als Bürokommunikationssoftware werden Anwendungen bezeichnet, die einen Mitarbeiter bei seiner Aufgabenerledigung in einem Büro durch verschiedene, kontextunabhängige, dialogorientierte Produkte unterstützen. BKS besteht dabei in der Regel aus mehreren Produkten, die nicht zwingend von einem Hersteller stammen müssen. Viele Hersteller bieten jedoch so genannte Office-Lösungen an, die eine Sammlung von benötigten Anwendungen beinhalten. Zu den Produkten gehören dabei Textverarbeitungsprogramme, Tabellenkalkulationssoftware, Zeichenprogramme, Präsentationssoftware, Programme für (elektronische) Post, Kalender und durch die Verbreitung und Informationsvielfalt des Internets auch ein Webbrowser. Die einzelnen Produkte innerhalb einer Office-Lösung nutzen oft produktspezifische Standards zur Ablage der mit ihnen erzeugten Dokumente oder entsprechende Schnittstellen, die auch Fremdsystemen (Workflowmanagementsysteme, Fachanwendungen, etc.) den Zugriff ermöglichen. Eine Auflistung der verbreitetsten Standards ist im erstellten „Software- und Kriterienkatalog“ [BHRV05] zu finden. Die Anforderungen an Bürokommunikationssoftware umfassen die Möglichkeit des Zugriffs über standardisierte Schnittstellen, die Möglichkeit der Nutzung auf allen in einer Verwaltung eingesetzten Betriebssystemen, eine gute Updateunterstützung für alle Arbeitsplatzrechner, auf denen die Software installiert ist und natürlich die verfügbaren Programmpakete. Verbreitete Office-Pakete sind StarOffice<sup>1</sup>, OpenOffice<sup>2</sup>, Koffice<sup>3</sup>, Microsoft Office<sup>4</sup> und AppleWorks<sup>5</sup>.

**Geographische Informationssysteme (GIS):** Geographische Informationssysteme bilden ein wesentliches Hilfsmittel für die Erfassung, Modellierung, Analyse und Visualisierung raumbezogener Daten und bestehen aus Hard- und Softwareanwendungen, deren Funktionen von der zugrunde liegenden Fachanwendung abhängig sind. GIS-Systeme bieten drei Sichten für die Arbeit mit geographischen Informationen: Geodatenbank-, Visualisierungs- und Verarbeitungs-Sicht, aus denen sich die Grundanforderung ergeben. Diese sind die Datenerfassung/Datenerstellung, Analyse, Visualisierung, Ausgabe/Verbreitung und die Datenkonvertierung. Bei geographischen Informationssystemen muss zwischen zwei Arten unterschieden werden: Anwendungslösungen und Serverlösungen. Anwendungslösungen bieten dem Nutzer die direkte Interaktion mit der auf ihrem Arbeitsplatzrechner installierten GIS-Software. Im Gegensatz dazu werden Serverlösungen genutzt, um dem Anwender raumbezogene Daten von einem zentralen Server über ein Frontend, was z.B. eine Applikation des Herstellers aber auch ein Webbrowser sein kann, zur Verfügung zu stellen. Eine Schwierigkeit bei der Anbindung von GIS-Systemen über standardisierte Schnittstellen ist die Vielfalt der Anwendungen. Hinzu kommt, dass fast jedes System sein eigenes, zumeist proprietäres Format zur Speicherung von Geodaten einsetzt. Eine Universallösung gibt es hier also nicht, da selbst innerhalb der öffentlichen Verwaltungen in Deutschland eine große Auswahl verschiedener Lösungen eingesetzt wird.

---

<sup>1</sup><http://de.sun.com/products/software/star/staroffice>

<sup>2</sup><http://www.openoffice.org>

<sup>3</sup><http://www.koffice.org>

<sup>4</sup><http://office.microsoft.com>

<sup>5</sup><http://www.apple.com/appleworks>

**Sicherheitskomponenten:** Die elektronische Umsetzung von Verwaltungsvorgängen innerhalb der IuK-Infrastruktur der öffentlichen Verwaltung erfordert die Beachtung verschiedener datenschutzrechtlicher Vorgaben unter Berücksichtigung der jeweiligen kommunikationswegspezifischen Anforderungen. Dies bedingt einerseits die Berücksichtigung von Aspekten der Datensicherheit (Verfügbarkeit, Datenintegrität, Verbindlichkeit und Vertraulichkeit) und des eigentlichen Datenschutzes (i.S. des Schutzes der Daten vor Missbrauch). Verfügbarkeit bezieht sich auf die Garantie, dass angeforderte Daten zum Zeitpunkt einer Anfrage bereitgestellt werden können. Die Sicherung der Datenintegrität dient dem Schutz der Daten vor ungewollten Modifikationen, wobei dies Veränderungen der Daten durch Hard- und Softwarefehler, gezielte Datenlöschung oder –veränderung durch Schadroutinen und gezielte Angriffe einschließt. Mit Verbindlichkeit werden die Authentizität von Daten, d.h. die Echtheit des Urhebers, und die Dokumentenintegrität, d.h. die Unverfälschtheit eines elektronischen Dokuments gegenüber seinem Original, sichergestellt. Das Vertrauen in die elektronische Kommunikation setzt neben dem Schutz der Verbindlichkeit durch Dokumentenintegrität den Schutz vor unberechtigter Kenntnisnahme und Vervielfältigung voraus. Bei der Untersuchung von Sicherheitskomponenten werden vor allem digitale Signaturen und Verschlüsselungsalgorithmen nebst zugehöriger Software und Zertifikaten betrachtet.

## 2.2 Kriterienkatalog

Im Kriterienkatalog wurden die untersuchten Softwaresysteme nach Kriterien aus fünf Kategorien (Bewertungsdimensionen) bewertet. Die Bewertung stellt die Grundlage für den später entwickelten Prototyp dar. Die erste Bewertungsdimension umfasst Kriterien bezüglich *Hersteller/Entwickler*. Hier werden die Aspekte *Lizenztyp, Verbreitung, Weiterentwicklung und Modifizierbarkeit* zur Einschätzung der Software genutzt. In der Bewertungsdimension *Systemumgebung* wurden unterstützte *Betriebssysteme*, eventuell notwendige *Laufzeitumgebungen, Schnittstellen* zur Verbindung mit anderen Systemen, der *Softwaretyp* und *Verteilungsmöglichkeiten* betrachtet. Die dritte Kategorie diente der Prüfung der untersuchten Software hinsichtlich *Standards*. Dazu zählen Standards nach *BGG §7 (Benachteiligungsverbot für Träger öffentlicher Gewalt)*, *Ergonomiestandards, unterstützte Dokumentenformate, XML- Standards* und *Standards bezüglich der Steuerung*. Für den Einsatz von Software in öffentlichen Verwaltungen sind Kriterien, die die *Sicherheit der Software* einordnen, besonders wichtig. In dieser Bewertungsdimension wurden dazu *Integration in bestehende Sicherheitsarchitekturen, Nutzerauthentifizierungs- und Nutzerautorisierungsmechanismen* sowie Einsatzmöglichkeiten von *Verschlüsselung und Signaturen* erfasst. Die letzte Kategorie umfasst eine Einordnung besonderer Eigenschaften (*Features*) der untersuchten Software. Dazu wurden *Internationalisierungsmöglichkeiten, Archivierung* und der *Austausch von Komponenten* der Software als Kriterien festgelegt. Der Kriterienkatalog der einzelnen Bestandteile des Systemkatalogs wird detailliert in [BHRV05] vorgestellt. Im Folgenden werden die einzelnen Teile deshalb nur kurz beschrieben.

**Dokumentenmanagementsysteme (DMS):** Bei der Erstellung von Bewertungskriterien für DMS ist zu berücksichtigen, dass es sich i.d.R. um fertige Softwarepakete handelt. Damit steht die Erweiterung oder Modifikation des DMS im Vergleich zur nahtlosen Integration in die bestehende IuK-Systemumgebung im Hintergrund.



**Bürokommunikationssoftware (BKS):** Bei der Erstellung von Bewertungskriterien für BKS ist zu berücksichtigen, dass es sich i.d.R. um fertige Softwarepakete handelt, welche zur täglichen Arbeit genutzt werden. Damit steht die Erweiterung oder Modifikation des BKS im Vergleich zur nahtlosen Integration in die bestehende IuK-Systemumgebung im Hintergrund.

**Geographische Informationssysteme (GIS):** Bei der Erarbeitung von Kriterien zur Auswahl zu nutzender GIS-Software bzw. entsprechender Basistechnologien und Werkzeuge sind die folgenden Hauptgesichtspunkte zu beachten. Zunächst ist die Frage zu klären, ob Richtlinien oder Empfehlungen zur Auswahl der Systeme existieren oder, ob ohne Einschränkungen entschieden werden kann. Weiter ist die Nutzung möglichst weit verbreiteter, offener Systeme notwendig und einhergehend mit der Entscheidung für den Einsatz freier Software. Die letzte zu klärende Frage ist, ob ein fertiges GIS-System oder ein offenes GIS-Framework genutzt werden soll. Zur konkreten Entscheidung über die Anbindung von GIS-Systemen ist zu klären, welche Dokumentenformate/Datenbanken/Datenformate genutzt werden sollen, welche Betriebssysteme unterstützt werden müssen, ob die Bibliotheken/APIs in der gewünschten Programmiersprache nutzbar sind, ob der Funktionsumfang ausreichend ist und die Stabilität der Implementierung des Produktes einhergehend mit der Weiterentwicklung.

**Sicherheitskomponenten:** Bei der Betrachtung verschiedener Komponenten zur Sicherstellung datenschutzrechtlicher und datensicherheitsrelevanter Vorgaben sind verschiedene Kriterien zu prüfen. Dazu gehören Grundsatzfragen zur unterstützten Systemumgebung, zum Lizenzmodell und zur Softwareweiterentwicklung, die kosten- und wartungsseitig Einfluss auf eine Entscheidung nehmen. In Bezug auf Sicherheitskomponenten sind folgende Kriterien zu berücksichtigen:

*Kriterium Standards:* Im Bezug auf die eingesetzten Standards müssen normative Vorgaben, wie die Konformität zum SigG und der SigV und der Einsatz von OSCI-konformen Protokollstandards sichergestellt werden. Konformität zum SigG und der SigV bedeutet, dass möglichst qualifizierte elektronische Signaturen unter Einsatz eines hierarchischen Zertifizierungssystems und asymmetrische Kryptographieverfahren genutzt werden. Beim Einsatz von OSCI-konformen Protokollstandards müssen die im OSCI-Transport, Teil A [OSC02] spezifizierten Protokollstandards zum sicheren Austausch elektronischer Dokumente vom eingesetzten Transportprotokoll unterstützt werden.

*Kriterium Features:* Hier ist eine Unterscheidung zwischen notwendigen Features und der Erhaltung der Flexibilität zu treffen. Notwendige Features umfassen eine Überprüfung, ob Anforderungen existieren, die die Nutzung eines speziellen Features erzwingen. Ist dies der Fall muss herausgefunden werden, ob dieses im Produkt enthalten sein soll oder individuell dazu programmiert werden kann. Die Erhaltung der Flexibilität umfasst die Prüfung, ob eingesetzte Features die Flexibilität (bspw. beim Einsatz bestimmter Komponenten) beeinträchtigen, was nicht der Fall sein darf.

## 2.3 Bewertung der Systeme

Das Ziel der Erstellung des Software- und Kriterienkatalogs bestand in der Erstellung einer Bewertungsmatrix, die Stärken und Schwächen der untersuchten Systeme herausstellt. Dabei wurden die Kriterien mit Ihrer jeweiligen Gewichtung für Entscheidungen der

späteren Realisierung genutzt. Die Bewertungsmatrix inklusive aller Informationen, die in den System- und Kriterienkatalog eingeflossen sind, ist in „Software- und Kriterienkatalog zu RAfEG - Referenzarchitektur für E-Government“ [BHRV05] zu finden.

Mit dem Ziel einer möglichst nahtlosen Integration in die bestehenden IT-Systemumgebungen wurden insbesondere offene Standards und weit verbreitete Schnittstellen der im Systemkatalog ausgewählten Produkte berücksichtigt. Im oben genannten ausführlichen Katalog werden zudem Ausschlusskriterien hinsichtlich der Verwendung von offenen Standards und Schnittstellen formuliert, deren Erfüllung zwingend mit der Integration eines Systems in die RAfEG-Systemarchitektur verbunden ist.

### **3 Entwicklung der Referenzarchitektur**

Die Architektur RAfEG stellt einen ganzheitlichen Ansatz dar, der viele wesentliche Aspekte des E-Governments, beginnend mit der formalen Beschreibung der fachlichen Zusammenhänge bis hin zur Entwicklung der Referenzarchitektur, basierend auf verteilt agierenden Softwarekomponenten zur Abarbeitung behördlicher Geschäftsprozesse, umfasst. RAfEG konzentriert sich dabei auf folgende Schwerpunkte:

- Entwicklung einer auf Open Source Software basierenden, an eine Vielzahl von Verwaltungsprozessen anpassbaren Referenzarchitektur,
- Entwurf von Konzepten zur effizienten Nutzung von heterogenen Systemen für interaktive Anwendungen im Bereich E-Government,
- Erstellung eines Baukastensystems mit notwendigen Modellen, Methoden und weiteren softwaretechnologischen Elementen wie Schnittstellen, Protokollen usw. für eine weitestgehend elektronische Realisierung von Verwaltungsprozessen,
- Selektion und Entwicklung von Architekturteilen, die durch eine formale (axiomatische oder automationsorientierte) Spezifikationen die für Verwaltungen typischen gesetzlichen Vorgaben im Prozessablauf berücksichtigen und
- Bereitstellung von Komponenten einer flexibel prozessunterstützenden Vorgangsteuerungssoftware.

Die Softwarearchitektur des RAfEG-Systems ist dabei als räumlich verteiltes, komponentenbasiertes Softwaresystem, das auf Open Source Software basiert und in einem heterogenen Systemumfeld einsetzbar ist, konzipiert.

Bei der Planung und Entwicklung von Softwarearchitekturen zur Implementierung bei der öffentlichen Hand sind teilweise strenge rechtliche Reglementierungen zu berücksichtigen. Dazu wurden in der Vergangenheit bereits Rahmenbedingungen geschaffen, die der Erkenntnis Rechnung tragen, dass sich eine gewünschte Interoperabilität zwischen heterogenen, verteilten (IT-)Komponenten nur durch eine entsprechende Standardisierung erreichen lässt. Unter dem Dach der Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung (KBSt) wurden Standards und Architekturen für E-Government-Anwendungen (SAGA) [Bun03] entwickelt. SAGA enthält entsprechende Empfehlungen hinsichtlich der Prozessmodellierung, der Schnittstellen, zu nutzender Techniken, der IT-Infrastruktur sowie der Datenmodellierung im Sinne eines Referenzmodells für offene, verteilte Datenverarbeitung (RM-ODP). Bei der Erarbeitung der RAfEG-Softwarearchitektur wurde auf Grundlage der Empfehlungen in

SAGA sehr hoher Wert auf eine komponentenbasierte Softwarearchitektur und die Trennung der Präsentations-, Funktions- und Datenlogik gelegt. Empfohlene Standards und Techniken konnten nach eingehender Prüfung der entsprechenden Alternativen bestätigt werden. Der Einsatz der offenen Standards XML [W3C07a] und XSLT [TL02] innerhalb der RAfEG Softwarearchitektur ist zweckmäßig. Die Integration empfohlener Komponenten (bspw. SAGA Basiskomponenten) und Standards (bspw. OSCI-A und -B [OSC01], XML-Projekte der Bundesverwaltung) ist innerhalb der vorgeschlagenen Architektur möglich, aber kein zwingender Bestandteil.

### 3.1 Softwarearchitektur-Komponente

Die Konzeption der Softwarearchitektur des RAfEG-Softwaresystems zur Abarbeitung von E-Government Prozessen umfasst die Planung der enthaltenen Softwarekomponenten und die funktionale Sicht des Gesamtsystems. Dies beinhaltet insbesondere die Zuordnung von Funktionalitäten zu einzelnen Komponenten, die statischen Beziehungen zwischen Komponenten sowie die dynamische Zusammenarbeit der Komponenten. In den weiteren Ausführungen werden die Begriffe RAfEG-Softwaresystem und RAfEG-System häufig genutzt. Als RAfEG-Softwaresystem wird dabei die Softwarearchitektur und die Implementierung bezeichnet, wogegen das RAfEG-System die Kombination aus Implementierung und zugrunde liegender Hardware bezeichnet. Das gesamte RAfEG-Softwaresystem ist so konzipiert, dass es in einzelne Komponenten aufgegliedert ist, die zum einen über mehrere Systeme verteilt ausgeführt und die zum anderen bei Bedarf ausgetauscht werden können. Die verteilte Ausführung ermöglicht dabei eine Lastverteilung und die Möglichkeit sicherheitskritische Komponenten auf speziell gesicherten Servern auszuführen. Die Möglichkeit des Austauschs einzelner Komponenten bedeutet ein hohes Maß an Flexibilität bei der Einführung des Systems in eine bestehende IT-Infrastruktur und ein leichtes Anpassen an sich ändernde Bedingungen, z. B. bei Austausch der Nutzerverwaltung in einer Behörde. Der Gesamtüberblick der Komponenten des RAfEG-Softwaresystems ist in Abbildung 1 dargestellt und wird in Abschnitt 4.3 detailliert erläutert.

Softwaresysteme zur elektronischen Unterstützung von Verwaltungsverfahren müssen sowohl die komplexen Abläufe innerhalb der am Verfahren beteiligten Institutionen als auch die Kommunikation zwischen prozessintegrierten externen Behörden mit eigenen Verwaltungsverfahren und externen Institutionen unterstützen. Daraus wurden ein Workflowsystem, ein Kommunikationssystem, eine zentrale Präsentationsschicht und ein Dokumentenmanagementsystem als essentielle Bestandteile eines solchen Softwaresystems abgeleitet. Das Dokumentenmanagementsystem wurde als Bestandteil des Backend-Systems geplant und ist in der Abbildung nicht dargestellt.

Im Unterschied zu herkömmlichen Systemen zur elektronischen Unterstützung von Verwaltungsverfahren berücksichtigt die RAfEG-Systemarchitektur:

- flexible Workflowinstanzen zur Modifikation der oft über Jahre laufenden Verwaltungsverfahren auch während der Abarbeitung von Verfahrensinstanzen,
- die Integration in die jeweils bestehende IT-Infrastruktur und Unterstützung herkömmlicher Kommunikationswege sowie
- die Portierbarkeit der Software und auch vorliegender Workflows auf andere Organisationseinheiten durch Nutzung einer Softwarearchitektur mit konsequentem Einsatz adaptiver Komponenten.

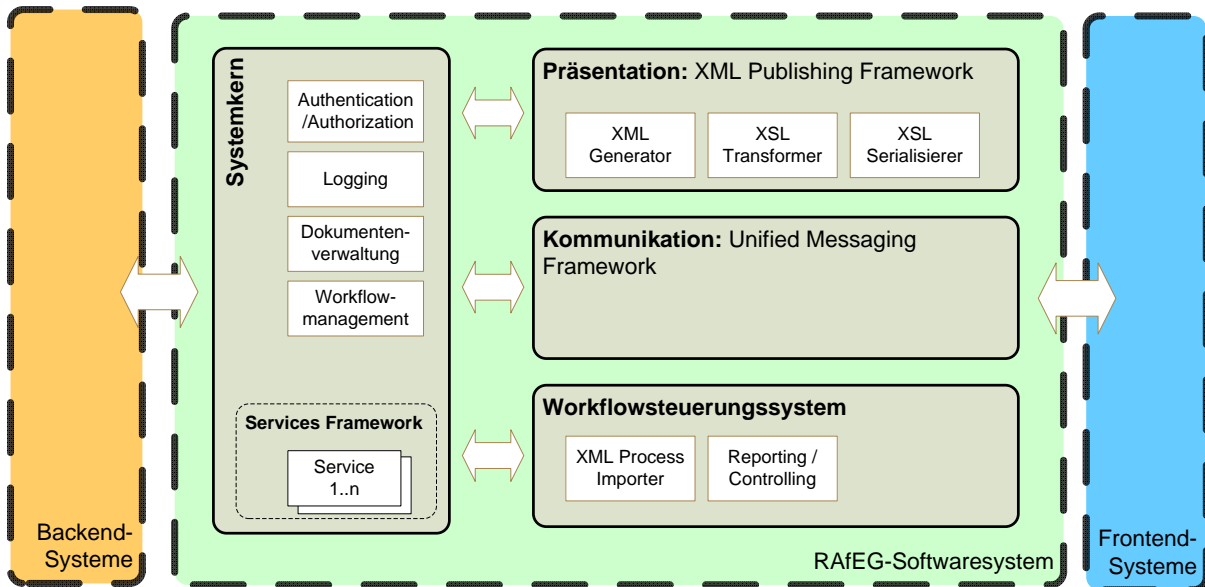


Abbildung 1: Überblick des RAfEG-Softwaresystems mit den einzelnen Baukastensystemkomponenten

Zentraler Bestandteil des RAfEG-Softwaresystems ist der Systemkern. Der Systemkern ist als Vermittlungsstelle zwischen Backend-Systemen und den RAfEG-Systemkomponenten Präsentations-, Kommunikations- und Workflowssteuerungssystemkomponente konzipiert. Das Backend-System umfasst zum einen Software von Drittanbietern und zum anderen selbst implementierte Zusatzsoftware, die nicht als zentraler Bestandteil des RAfEG-Softwaresystems geplant wurde. Backend-Systeme sind z.B. das Dokumentenmanagementsystem, ein Datenbanksystem für die Speicherung von Konfigurations- und Log-Daten und ein Nutzerverwaltungssystem, z.B. LDAP. Zusätzlich vermittelt der Systemkern zwischen den einzelnen Systemkomponenten, da diese bei der verteilten Abarbeitung von Workflows (Workflowsteuerungssystem ↔ Kommunikation) und bei der Interaktion von Nutzern mit dem System (Präsentation ↔ Workflowsteuerungssystem) zusammenarbeiten. Das erstellte Architekturkonzept sieht weiterhin vor, dass die Komponenten auf verschiedenen Systemen abarbeitbar sein sollen, was eine Implementierung von Kommunikationsmethoden über ein Netzwerk erforderlich macht. Wichtig ist in diesem Zusammenhang, dass innerhalb der Workflowssteuerungssystemkomponente das Workflowmanagementsystem (WfMS) als Bibliothek genutzt wird. Der Zugriff wird über generische Schnittstellen ermöglicht. Die Implementierung der Workflowsteuerungssystemkomponente wird in Abschnitt 4.3 vorgestellt.

Nach der Einführung in die statische Struktur und den Aufbau der RAfEG-Systemarchitektur wird nun auf die Systemarchitektur für die verteilte Workflowabarbeitung eingegangen. In dieser Darstellung wird eine Instanz, d.h. ein durch ein WfMS abgearbeiteter Workflow, als Prozess bezeichnet. Das Protokoll für die verteilte Arbeitung von organisationsübergreifenden Verwaltungsvorgängen wurde in Anlehnung an Spezifikationen und Standards der Workflow Management Coalition (WfMC) [WFM] erstellt. Es basiert auf dem Ansatz, dass die kommunizierenden Systeme jeweils als Client (Dienstnutzer) und Server (Dienstanbieter) fungieren können und auch müssen. Im konkreten Fall von RAfEG bedeutet dies, dass ein System (Client) Dienste (z. B. Start eines neuen Prozesses, Rückgabe von Variablen bei Prozessende) eines anderen Systems (Server) nutzt.

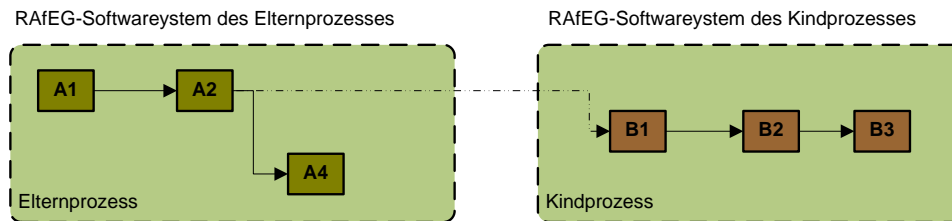


Abbildung 2: Modell der verketteten Prozesse. Die Aufgaben A1, A2 und A4 gehören zum Elternprozess und die Aufgaben B1 bis B3 zum Kindprozess.

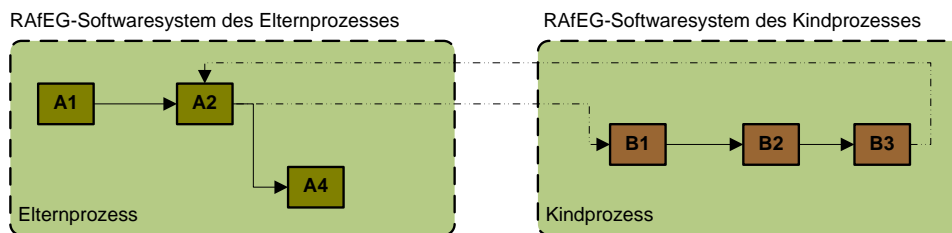


Abbildung 3: Modell der hierarchischen Unterprozesse. Die Aufgaben A1, A2 und A4 gehören zum Elternprozess und die Aufgaben B1 bis B3 zum Kindprozess.

Für die verteilte Abarbeitung nehmen dabei die beteiligten Systeme sowohl die Rolle als Client, als auch als Server ein. Durch die WfMC werden vier Szenarien für eine verteilte Workflowabarbeitung (Remote-Prozessabarbeitung) definiert [Hol95]. Zwei dieser Szenarien für Remote-Prozessabarbeitung sind für RAfEG von Bedeutung und wurden deshalb, in angepasster Form, in das System integriert. Das sind zum einen das „Modell der verketteten Prozesse“ und zum anderen das „Modell der hierarchischen Unterprozesse“. Das Modell der verketteten Prozesse besteht aus zwei voneinander abhängigen Prozessen. Abhängig bedeutet in diesem Kontext, dass der erste Prozess (im Weiteren als Elternprozess bezeichnet) den Zweiten (Kindprozess) während seiner eigenen Abarbeitung entfernt (remote) startet. Nach dem erfolgreichen Start des Kindprozesses fährt der Elternprozess mit seiner Ausführung fort und die beiden Prozesse werden ab diesem Zeitpunkt parallel zueinander weiter abgearbeitet, siehe Abbildung 2.

In diesem Szenario ist es dem Elternprozess zwar möglich, dem Kindprozess Eingabeparameter mitzugeben, jedoch können keine Werte zurückgegeben werden. Sollen Werte zurückgegeben werden bzw. ist der Ausgang des externen Teilprozesses des Verwaltungsprozesses für die weitere Abarbeitung von Bedeutung, wird das Modell der hierarchischen Unterprozesse genutzt. Hier gibt es zwei Kommunikationspunkte zwischen dem Eltern- und dem Kindprozess. Abbildung 3 illustriert dieses Szenario an einem einfachen Beispiel.

Um die Details des Kommunikationsprotokolls vom Anwendungsprogrammierer zu verstecken, wurde ein Ansatz gewählt, in dem die externe Abarbeitung völlig transparent automatisiert abläuft [Kun07]. Beachtet werden muss nur, dass der Prozessentwickler (Modellierer) den Workflow so erstellt, dass der Workflow den Namen (bei der Modellierung vorerst nur generisch) des entfernten Systems kennt, das Modell ausgewählt wird und die Ein- und Ausgabevariablen definiert werden. Dieses Prinzip wird ausführlich in [BKR05] und [BKR06] vorgestellt. Die an einer verteilten Workflowabarbeitung beteiligten RAfEG-Softwaresysteme – genauer die Workflowsteuerungssystemkomponenten

Tabelle 1: Schnittstellen der Server für die verteilte Workflowabarbeitung.

Methoden	Bereitgestellt von	Genutzt von
Start eines Prozesses ( <i>startProc</i> )	RAfEG-Softwaresystem des Kindprozesses	RAfEG-Softwaresystem des Elternprozesses
Information über Zustands- wechsel des Kindprozesses ( <i>informProcessStateChanged</i> )	RAfEG-Softwaresystem des Elternprozesses	RAfEG-Softwaresystem des Kindprozesses
Information über Zustands- wechsel einer Aufgabe des Kindprozesses ( <i>informActivityStateChanged</i> )	RAfEG-Softwaresystem des Elternprozesses	RAfEG-Softwaresystem des Kindprozesses
Information über Beendigung des Kindprozesses ( <i>informCompleted</i> )	RAfEG-Softwaresystem des Elternprozesses	RAfEG-Softwaresystem des Kindprozesses

der RAfEG-Softwaresysteme – müssen vier Methoden bereitstellen, um eine verteilte Abarbeitung zu ermöglichen, siehe Tabelle 1. Die in den letzten drei Zeilen grau unterlegten Methoden werden nur im Modell der hierarchischen Unterprozesse benötigt und sind für den Fall eines verketteten Prozesses irrelevant.

Die Parameter der einzelnen Methoden bestehen mindestens aus einem Identifikator des Senders und einem des Empfängers. Jedes System besitzt dazu genau einen, global eindeutigen Grundidentifikator (z.B. die IP-Adresse) und benutzt diesen mit zusätzlichen Informationen über den zugehörigen Prozess als Senderidentifikator, wenn es die Rolle des Client einnimmt und als Empfängeridentifikator, wenn es die Rolle des Servers einnimmt. Diese Identifikatoren sind für die Überprüfung, ob die Kommunikationspartner zueinander passen, notwendig und werden bei Start einer externen Prozessabarbeitung erzeugt. Zusätzlich besitzen die einzelnen Methoden die folgenden Parameter:

Die Methode *startProc* besitzt als zusätzliche Parameter

- den eindeutigen Prozessidentifikator des Elternprozesses des rufenden Systems, um eine Zuordnung von Daten der Kindprozesse zu einem Elternprozess zu gewährleisten,
- die Prozessdaten (workfloweigene Variablen, Dokumente und Aktennotizen) mit ihren Modi (Eingabe-, Ausgabe- oder Ein/Ausgabedaten) und
- die Angabe, ob es sich um einen verketteten Prozess oder einen hierarchischen Unterprozess handelt.

Die Methode *informProcessStateChanged* besitzt als zusätzliche Parameter

- den vorherigen Zustand des Kindprozesses (z.B. laufend, unterbrochen, abgeschlossen, ...),
- den aktuellen Zustand des Kindprozesses und
- den Zeitpunkt der Zustandsänderung.

Die Methode *informActivityStateChanged* besitzt als zusätzliche Parameter

- einen innerhalb des externen Prozesses eindeutigen Aufgabenidentifikator,

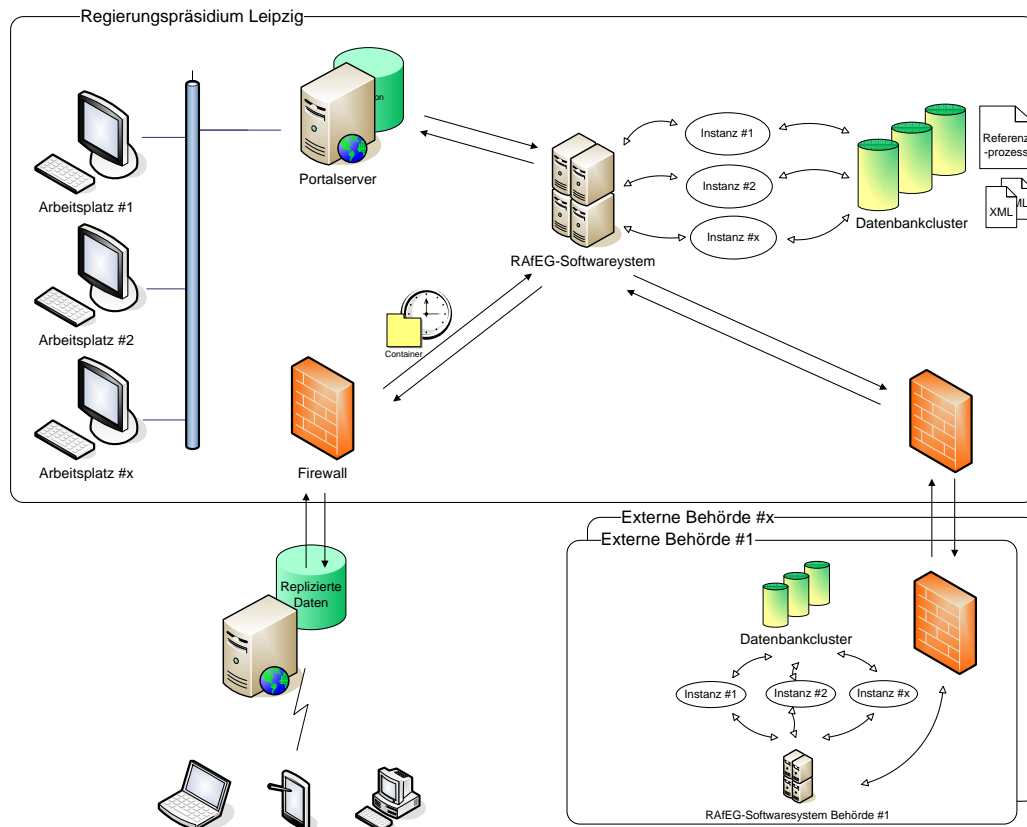


Abbildung 4: Sicht auf das RAFEG-System bestehend aus Arbeitsplatzrechnern, Servern und den Schnittstellen zwischen den einzelnen Hardwarekomponenten.

- den Namen der Aufgabe des Kindprozesses,
- den vorherigen Zustand der Aufgabe des Kindprozesses,
- den aktuellen Zustand der Aufgabe des Kindprozesses und
- den Zeitpunkt der Zustandsänderung.

Die Methode *informCompleted* besitzt als zusätzlichen Parameter

- die Prozessdaten (workfloweigene Variablen, Dokumente und Aktennotizen).

### 3.2 Infrastrukturarchitektur-Komponente/Integration zu RAFEG

Innerhalb der Infrastrukturarchitektur-Komponente wird die operationale Sicht auf das System beschrieben. Dabei wird der Bogen zwischen dem technischen System (mit Hardware, Plattformen, Verbindungen), der Platzierung der Softwarekomponenten im Rahmen des technischen Systems, der Konfiguration des Systems und dem Management des Systems (Kapazitätsplanung, Softwareverteilung, Datensicherung und Wiederanlauf) gespannt. Für den Aufbau des technischen Systems wurden zuerst die typischen hardwaretechnischen Gegebenheiten, wie sie in Behörden anzutreffen sind, erfasst und darauf aufbauend die Infrastrukturarchitektur des RAFEG-Systems erarbeitet. Einen Überblick gibt Abbildung 4.

In den folgenden Abschnitten werden die einzelnen Teilkomponenten der hardwaretechnischen Voraussetzungen eingeführt und sich daraus ergebende Anforderungen an die Infrastrukturarchitektur herausgestellt. Als erstes sollen dazu die Arbeitsplatzrechner analysiert werden. Im Regierungspräsidium Leipzig sind als Arbeitsplatzrechner Standard PCs eingesetzt, die mit dem Windows Betriebssystem von Microsoft ausgestattet sind. Weiterhin werden für die Anwendungen für Bürokommunikationssoftware zumeist Microsoft Lösungen eingesetzt. Die angestrebte Plattformunabhängigkeit der RAfEG-Systemarchitektur kann hier durch den Einsatz eines Webbrowsers erreicht werden, da dieser nicht nur für Microsoft Windows Systeme verfügbar ist. Zusätzlich ist eine strikte Trennung von Präsentation und Anwendungslogik notwendig. Die Anwendungslogik ist dabei komplett aus den Arbeitsplatzrechnern auszulagern. Verbunden sind die Arbeitsplatzrechner mit einem Portalserver. Hinter diesem Begriff verbirgt sich ein Server, der für die in den Arbeitsplatzrechnern zu nutzenden Webbrowser die anzuzeigenden Daten erzeugt. Die Anforderungen an den Portalserver ist vor allem die schnelle Beantwortung von Anfragen durch Nutzer. Dazu ist es möglich nicht nur einen, sondern mehrere Portalserver in die Infrastruktur der Behörde zu integrieren und eine Lastverteilung bei einer hohen Anfragedichte zu bieten. Das Vorhandensein mehrerer Portalserver ist dabei für den Nutzer transparent zu gestalten. Die Anfragen des Nutzers an den Portalserver werden von diesem an das RAfEG-System weitergeleitet. Als Hardwareschnittstelle ist hier Ethernet bzw. Fast Ethernet anzuraten, was behördenintern auch meist als internes Kommunikationsnetzwerk verwendet wird. Die Komponente des RAfEG-Softwaresystems, die auf dem Portalserver ausgeführt wird, ist Teil des Baukastensystems und kann somit auch innerhalb des Servers/Clusters des RAfEG-Systems ausgeführt werden. In diesem Fall entfällt der zusätzliche Portalserver. Das RAfEG-Softwaresystem seinerseits stellt die Anwendungslogik bereit und verrichtet daher den Großteil der Arbeit, d.h. Datenbankabfragen, Nutzerprüfung, Zusammenarbeit mit dem WfMS, usw. Eine Möglichkeit für kleine Behörden ist hier, das RAfEG-Softwaresystem auf einem einzigen Server auszuführen. Dieses Vorgehen kann allerdings bei großen Behörden mit intensiver Nutzung des RAfEG-Systems schnell an seine Grenzen stoßen. Die Konzeption des RAfEG-Softwaresystems als Baukastensystem ermöglicht hier die Verteilung auf mehrere Server (Clusterung).

Zur Unterstützung des RAfEG-Softwaresystems wird ein Datenbankcluster verwendet, in dem zum einen Konfigurationsinformationen des Systems selbst und zum anderen die Daten des Workflowmanagementsystems abgelegt werden. Dabei wird der Großteil der Zugriffe auf das Datenbanksystem für Abfragen von Workflowinformationen genutzt. Hierbei ist anzumerken, dass jede Aktion eines Nutzers eine Interaktion mit dem WfMS darstellt, was eine sehr große Last des Datenbanksystems bedeutet. Deshalb ist es möglich das Datenbanksystem nicht nur auf einem Server auszuführen, sondern auch einen Datenbankcluster einzusetzen, auf dem die Daten repliziert vorliegen. Neben der Lastverteilung ist dadurch zusätzlich eine Sicherung von Daten gegen Ausfall eines Servers des Datenbankclusters gegeben.

Für die verteilte Abarbeitung von Verwaltungsprozessen ist es notwendig, dass die beteiligten Behörden – genauer die RAfEG-Softwaresysteme der beteiligten Behörden – miteinander kommunizieren können. Dabei ist die Architektur so aufgebaut, dass in jeder Behörde ein RAfEG-System läuft. Die Systeme sind über ein Netzwerk miteinander verbunden. Innerhalb Sachsens ist dies der Sächsische InfoHighway [Lei07]. An den InfoHighway sind alle 850 Landes-Behörden mit über 70.000 Beschäftigten über Standleitungen angeschlossen. Durch die Anbindung an dieses Hochgeschwindigkeitsnetz ist es möglich, schnell große Mengen an Daten zwischen den Behörden zu übertragen, was für externe Teile von



großen Verwaltungsvorgängen von zentraler Bedeutung ist. Die Firewalls zwischen Behörden und dem InfoHighway sind meist sehr restriktiv eingestellt, was die Möglichkeiten für den Datentransport sehr einschränkt. Eine Lösung unter Nutzung von HTTP (bzw. SOAP [W3C07b] über HTTP) ist hier zu bevorzugen – HTTP-Verbindungen sind in der Regel erlaubt – und wurde deshalb implementiert. Informationen dazu sind in Abschnitt 4.4 zu finden.

Für die Interaktion von externen Nutzern – in Abbildung 4 unten links dargestellt –, was z.B. beim Planfeststellungsverfahren durch Einwände von Bürgern gegen Bauvorhaben geschehen kann, müssen Daten über öffentliche Netze übertragen werden. Öffentliche Netze sind dabei solche, die nicht unter der Kontrolle der Behörde (das interne Netz) oder vertrauenswürdiger Organisationen (der sächsische InfoHighway) stehen. Hier können im schlimmsten Fall Daten eingesehen, manipuliert oder auch ein Server direkt angegriffen werden. Zu diesem Zweck ist es notwendig, für den öffentlichen Zugriff nur Kommunikation von nichtvertraulichen Daten zwischen externem System und dem RAfEG-Softwaresystem einer Behörde zuzulassen. Hier werden außerhalb der Behörde spezielle Server installiert. Diese müssen dabei nicht so leistungsstark sein, wie das System in einer Behörde, was dazu führt, dass dies i.d.R. genau ein Rechner ist, auf dem alle benötigten Komponenten (ein im Funktionsumfang eingeschränkter Portalserver, ein reduziertes RAfEG-Softwaresystem, ein Desktopdatenbanksystem) ausgeführt werden. Die Teile eines Verfahrens, die von einem behördenexternen Nutzer beeinflusst werden können, werden bei ihrer Ausführung an den externen Server gesendet. Diese beinhalten i.d.R. nur sehr wenige Workflowdaten. Der Nutzer arbeitet dabei auf einer Kopie der eigentlichen Workflowdaten des Gesamtsystems. Ist der externe Teil mit Nutzerinteraktion abgearbeitet (z.B. weil eine Frist beendet ist) werden die geänderten Daten an das System der Behörde zurückgesendet. Als wichtigstes Merkmal des externen Servers ist festzuhalten, dass der Aufruf von Prozessen des behördeninternen RAfEG-Softwaresystems vom externen Server verboten ist.

## 4 Prototypische Umsetzung (Implementierung)

Die prototypische Implementierung, der in Abschnitt 3 erstellten Konzepte und Architekturen, wurde mit den Zielen der Abarbeitung typischer Prozesse des E-Government, der Erweiterbarkeit, und der ausschließlichen Verwendung von Open Source Software umgesetzt. Durch das entstandene Baukastensystem ist es zudem möglich, Komponenten auszutauschen und damit das RAfEG-Softwaresystem an sich ändernde Bedürfnisse anzupassen. Da der Einsatz des RAfEG-Softwaresystems im Bereich des E-Government mit der Bearbeitung von sicherheitskritischen Daten verknüpft ist und speziell im Hinblick auf die verteilte Abarbeitung von Workflows auch hohe Risiken beim Versenden der Daten entstehen, wurde bei der Implementierung insbesondere auf Sicherheitsmechanismen Wert gelegt.

### 4.1 Protokoll mit Schnittstellen

Die prototypische Implementierung des Protokolls zur verteilten Abarbeitung von Workflows erfolgte zunächst unter Nutzung von RMI (Remote Method Invocation) [Gro01] als Kommunikationsprotokoll. Obwohl diese Realisierung noch ohne Sicherheitsaspekte durchgeführt wurde, bildet sie die Grundlage für die später erweiterte Version, in der OSCl-Transport für die sichere und nachvollziehbare Kommunikation eingesetzt wird.

Die Abarbeitung von Prozessen basiert auf der Abarbeitung von hierarchischen Workflows. Die Workflows sind dabei in AML (Aris Modelling Language) [IDS] modellierte Verwaltungsverfahren, die mittels einer erstellten externen RAfEG-Komponente für die Abarbeitung umgewandelt werden. Die Umwandlung erfolgt dabei von AML nach XPDL (XML Process Definition Language) [Nor02] und ist in [BDR06] detailliert dargestellt. XPDL wird von den im RAfEG-Softwaresystem eingesetzten Workflowmanagementsystemen benötigt, um die modellierten Prozesse abarbeiten zu können. Die Workflows sind dabei aus zwei Gründen hierarchisch aufgebaut. Zum einen ist der hierarchische Aufbau notwendig, da die Modellierung der Verwaltungsvorgänge in einem Schichtenmodell durch das Institut für Wirtschaftsinformatik (IWi) Saarbrücken erfolgte. Hier wurden die Verwaltungsvorgänge in ein 5-Schichtenmodell gepackt, das ausgehend vom Planfeststellungsverfahren als eine einzige Aufgabe (Schicht 1), bis hin zu Einzelaufgaben für einen Bearbeiter (Schicht 5) durchmodelliert ist. Dieses erlaubt neben der Möglichkeit von Tests einzelner Verfahrensteile vor allem den einfachen Austausch einzelner Teile eines Verfahrens. Zum anderen geschieht die externe Abarbeitung von Teilen eines Workflows über die externe Abarbeitung eines Unterprozesses (Subflow), was für die Realisierung der Schnittstelle den wichtigeren Teil ausmacht. Dieser Subflow wird dabei als eine Teilaufgabe des Workflows mit allen Ein- und Ausgabeparametern definiert, was den Austausch von externen Teilen bei einer späteren Neumodellierung stark erleichtert.

Der grundlegende Ablauf einer entfernten Prozessabarbeitung ist in Abbildung 5 dargestellt. Die Verwaltungsvorgänge sollen, wie bereits erwähnt, als Workflows in XPDL vorliegen und werden vom WfMS in der Workflowsteuerungssystemkomponente abgearbeitet. Ein einzelner Workflow besteht dabei aus Aufgaben. Hinter einer Aufgabe kann sich dabei eine durch einen Anwender ausführbare oder eine vom System automatisch abgearbeitete Aktion verbergen. Eine Aufgabe mit Nutzerinteraktion besitzt dabei zumeist Dokumente, die der Nutzer bearbeiten muss. Innerhalb des RAfEG-Softwaresystems werden diese Dokumente für jeden Prozess separat in einer Prozessmappe [KRS07a] abgelegt. Die Prozessmappe ist dabei ein im Dokumentenmanagementsystem abgelegter Ordner, der alle prozessrelevanten Daten (Dokumente, Aktennotizen, usw.) enthält und auf den durch die Workflowsteuerungssystemkomponente über den Prozessidentifikator zugegriffen werden kann. Externe Teilvorgänge werden als Unterprozesse in die Workflowbeschreibung integriert. Eine Aufgabe, die für eine externe Ausführung bestimmt ist, kann in XPDL also als automatisierte Aufgabe modelliert werden. Automatisiert bedeutet in diesem Fall, dass der Quellcode ohne Nutzerinteraktion vom Workflowmanagementsystem selbst abgearbeitet wird.

Als mögliche WfMS wurden Enhydra Shark [Enh05] und WfMOpen [LSW13] eingesetzt. Im Falle von Enhydra Shark und WfMOpen können verschiedene Programmiersprachen für diese automatisierten Aufgaben verwendet werden. Da das Gesamtprojekt in Java implementiert ist, wurde auch hier Java verwendet. Der auszuführende Quellcode der Schnittstelle ist fest programmiert, wobei das zu verwendende Kommunikationsmodul ausgetauscht werden kann. Die eigentliche Abarbeitung wird durch die, für jeden Prozess unterschiedlichen, modellierten Ein- und Ausgabeparameter auf spezielle Bedürfnisse angepasst. Die an der Kommunikation während der Ausführung des externen Prozesses beteiligten und in Abbildung 5 dargestellten Komponenten sind: ein Anwendungsagent (dieser entspricht dem Programm hinter der eben erläuterten automatisierten Aufgabe), einem Beobachter, einem Betriebsobjekt und einem Instanzobjekt. Die letztgenannten drei Objekte sind dabei in Enterprise JavaBeans enthalten (*BeobachterServerBean* und

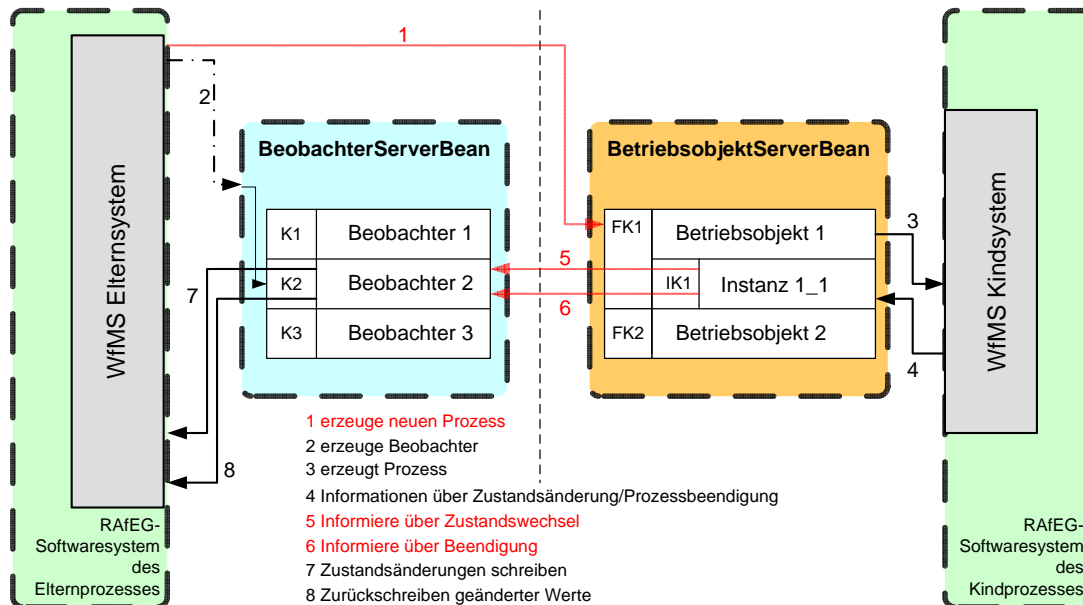


Abbildung 5: Kommunikationsarchitektur für die verteilte Abarbeitung von Verwaltungsvorgängen.

*BetriebsobjektServerBean*), um entfernten Systemen Zugriff zu erlauben und die in Abschnitt 3.1 vorgestellten Schnittstellen verfügbar zu machen.

**Anwendungsagent:** Der Anwendungsagent ist das oben erwähnte Codefragment, das in der Aufgabendefinition des externen Prozesses als automatische Aufgabe ausgeführt wird. Der ausgeführte Code ist dafür zuständig, den im modellierten Workflow mit Namen angegebenen externen Prozess zu starten. Dazu liest er die modellierten Ein- und Ausgabeparameter und stellt die zu versendenden Daten zusammen. Diese Daten umfassen:

- workfloweigene Variablen, z.B. Variablen aufgrund derer im Kindprozess Entscheidungen bezüglich der Abarbeitung getroffen werden können oder Variablen, die Informationen über Deadlines enthalten,
- Dokumente, wie z.B. Formulare, Anträge, u.Ä. und
- Aktennotizen, die Zusatzinformation bezüglich der Abarbeitung bereits abgeschlossener Aufgaben enthalten können.

Nach der Zusammenstellung der Daten kontaktiert der Anwendungsagent das in der Prozessbeschreibung angegebene Betriebsobjekt (Methode *startProc*). Dies wird über einen fest modellierten Namen und einer Adresszuordnungsdatei, die dynamisch änderbar ist, geregelt. Nachdem er die Anfrage zum Erzeugen des neuen Kindprozesses gestellt und die Daten gesendet hat, beendet er sich im Falle eines verketteten Prozesses und der Elternprozess läuft weiter. Im Falle eines hierarchischen Unterprozesses, startet er stattdessen einen Beobachter, der später noch erläutert wird, und eine Warteaktivität im Prozess, die später vom Beobachter beendet wird.

**Betriebsobjekt:** Ein Betriebsobjekt repräsentiert einen extern abarbeitbaren Prozess. Die Workflowsteuerungssystemkomponente jedes RAfEG-Softwaresystems startet während des Hochfahrens des RAfEG-Softwaresystems über eine Konfigurationsdatei für jeden von extern zugreifbaren Prozess ein Betriebsobjekt. Diese Objekte werden in der BetriebsobjektServerBean verwaltet und wenn eine Anfrage zum Start von einem Anwendungsagenten gestellt wird, wird durch das Betriebsobjekt der entsprechende Prozess im der Workflowsteuerungssystemkomponente zugrunde liegenden WfMS gestartet. Neben dem Start werden die Eingabeparameter gesetzt und Dokumente und Aktennotizen für den Prozess zugreifbar (in seiner Prozessmappe) abgelegt. Im Falle eines verketteten Prozesses ist damit die Aufgabe des Betriebsobjekts abgeschlossen. Wird jedoch ein Unterprozess vom Betriebsobjekt repräsentiert, wird zusätzlich ein Instanzobjekt angelegt.

**Instanzobjekt:** Jedes erzeugte Instanzobjekt ist direkt mit dem WfMS der startenden Workflowsteuerungssystemkomponente verbunden und erhält Informationen über Änderungen während der Abarbeitung des zugehörigen Prozesses. Diese Änderungen können dabei Zustandsänderungen oder die Beendigung des Prozesses sein. Zustandsänderungen werden über die Methoden *informProcessStatechanged* und *informActivityStatechanged* an den Beobachter dieses Prozesses weitergegeben. Wird der Prozess beendet, werden die zurückzusendenden Daten zusammengestellt und der Beobachter über die Methode *informCompleted* darüber informiert. Zusätzlich erhält er dabei die Rückgabedaten. Das Instanzobjekt wird nach erfolgreicher Übermittlung der Beendigungsnachricht entfernt.

**Beobachter:** Der Beobachter wartet auf Informationen durch ein Instanzobjekt betreffs Zustandsänderungen und Beendigung des zugehörigen externen Kindprozesses. Im Falle von Zustandsänderungen werden diese in einer Historie gespeichert, um einem Administrator die Möglichkeit zu bieten, Informationen über den Zustand des Kindprozesses einzusehen. Im Falle der Beendigung des Kindprozesses erhält der Beobachter zusätzlich die Rückgabedaten des Kindprozesses vom Instanzobjekt. Diese Rückgabedaten werden in die Warteaktivität des Elternsystems geschrieben und diese wird beendet, wodurch die Abarbeitung des Elternprozesses fortschreiten kann und die geänderten Werte den Folgeaufgaben zur Verfügung stehen. Dokumente und Aktennotizen werden zusätzlich in die Prozessmappe des Elternprozesses geschrieben, aus der auch schon beim Start die Eingabeinformationen des externen Prozesses entnommen wurden. Ist die Aktion bei Beendigung des externen Prozesses an den Elternprozess erfolgreich abgeschlossen worden, wird der Beobachter aus der BeobachterServerBean entfernt.

## 4.2 Prototypische Realisierung ohne Sicherheitsaspekte

Das RAfEG-Softwaresystem beinhaltet die grundlegenden Komponenten, die zur elektronischen Abarbeitung von Verwaltungsvorgängen notwendig sind. Die Komponenten werden in der prototypischen Realisierung innerhalb eines Application Servers in Form von Enterprise JavaBeans ausgeführt. Als Application Server wird der JBoss AS [JBo07] verwendet. Die Modellierung einzelner Komponenten als Enterprise JavaBeans [Sta04, Mon04] ermöglicht es, zusammenarbeitende Komponenten entweder auf genau einem Application Server oder transparent auf mehreren Servern verteilt auszuführen (Clusterung). Neben der Möglichkeit der Verteilung von Komponenten, bietet die Nutzung von JBoss AS weiterhin die Unterstützung von Sicherheitsrichtlinien für die Komponenten. So ist es nicht nur möglich die Komponenten vor fremdem Zugriff zu schützen, sondern

zusätzlich einzelne Funktionen nur bestimmten Nutzern zu Verfügung zu stellen. Durch die Verteilung der Komponenten ist es möglich eine Lastverteilung zu realisieren, was besonders in größeren Behörden zum Tragen kommt, in denen sehr viele Nutzer auf das System zugreifen. Für die Kapselung der einzelnen Bausteine des Baukastensystems wurde jede Komponente in ein so genanntes Enterprise Application Archive (EAR) gepackt und ist unabhängig von den anderen Komponenten auf einem Server mit installiertem JBoss AS ausführbar. Die einzelnen Komponenten und Ihre Hauptbestandteile – jeweils als eigenständige Enterprise JavaBean realisiert – sind in Tabelle 2 dargestellt.

Tabelle 2: Kernkomponenten mit Hauptbestandteilen und deren Funktionalitäten.

<b>Baustein</b>	<b>Kernobjekt</b>	<b>Funktionalität</b>
Systemkern	Config	Konfiguration des Systems, z.B. festlegen des genutzten Datenbanksystems, des Authentifizierungssystems usw.
	Filestore	Schnittstelle zum Dokumentenmanagementsystem, ermöglicht den Zugriff der anderen RAfEG-Komponenten auf das DMS.
	Geodatabase	Zugriff auf eine Geodatenbank, die für eine Umkreissuche bei der Auswahl von betroffenen Trägern öffentlicher Belange beim Planfeststellungsverfahren genutzt werden kann.
	Users	Funktionen für die Nutzerverwaltung bzw. Schnittstellen zum Authentifizierungssystem des Backend-Systems.
Präsentation	Cocoon-Generation	Generatormethoden für Apache Cocoon, die aus der zurückgegebenen Datenrepräsentation der Anfragen an die anderen Komponenten eine Zwischendarstellung mit Formatierungsanweisungen erzeugt.
	Cocoon-Transformation	Transformationsmethoden, die aus der von den Generatormethoden erzeugten Zwischendarstellung das eigentliche Ausgabeformat erzeugen (z.B. HTML oder PDF).
Kommunikation	Communicator-Supplier	Kommunikator für den Server. Dieser stellt Methoden für das Empfangen von Nachrichten bereit. Der Kommunikator kann dabei verschiedene Transportprotokolle für das Empfangen von Nachrichten verwenden.
	Communicator-Client	Kommunikator für den Client. Stellt den anderen RAfEG-Komponenten Methoden zur Verfügung, über die Nachrichten an einen Server über ein beliebiges Transportprotokoll versendet werden können.

*Tabelle wird auf der nächsten Seite fortgesetzt*

Fortsetzung der Tabelle		
Baustein	Kernobjekt	Funktionalität
Workflowsteuerungssystem	WfMSServer	Zentrale Komponente des Workflowsteuerungssystems, die die generischen Schnittstellen für den Zugriff auf das, als Bibliothek genutzte, WfMS bereitstellt. Die anderen Komponenten kommunizieren mit einer Instanz dieser Serverklasse, um den tatsächlich eingesetzten WfMS-spezifischen Server zu verstecken.
	SharkWfMSServer	Implementierung von WfMSServer für Shark.
	WfmOpenWfMS-Server	Implementierung von WfMSServer für WfmOpen.
	FactoryServer	Die Realisierung des Betriebsobjektsservers für die verteilte Abarbeitung von Workflows.
	ObserverServer	Die Realisierung des Beobachterservers für die verteilte Abarbeitung von Workflows.
	Workflow-Communicator	Spezielle Implementierung des Kommunikators für den Einsatz mit Workflowmanagementsystemen (enthält eine Implementierung die den Communicator-Supplier und auch den Communicator-Client nutzt).

### 4.3 Baukastensysteme

Der Aufbau des RAfEG-Softwaresystems, wie er in Abbildung 1 dargestellt ist, zeigt die grundlegende Struktur des Baukastensystems. Die Konzeption der RAfEG-Systemarchitektur wurde dabei so angelegt, dass die einzelnen Komponenten austauschbar sind. Die Komponenten werden in diesem Abschnitt detailliert vorgestellt. Die Frontend- und Backendsysteme sind an das RAfEG-Softwaresystem über offene Schnittstellen angeschlossen. Sie gehören nicht zum erstellten Baukastensystem, obwohl diese zum Teil auch selbst implementiert wurden. Auf diese Systeme wird am Ende dieses Abschnitts eingegangen.

**Systemkern:** Der Systemkern fasst Basisfunktionen zusammen, die für alle Systemkomponenten notwendig sind. Hierzu zählt die Bereitstellung von Zugriffsmechanismen auf die, in der Abbildung als Backend-Systeme (nachgeschaltete Komponenten) bezeichnete Software über freie Protokolle (z.B. LDAP für den Zugriff auf die Nutzerverwaltungssoftware). Weiterhin ist hierin die Verwaltung und Konfiguration der Systemkomponenten sowie die Berechtigungsverwaltung für das gesamte System enthalten. Der Systemkern nutzt dabei einerseits innerhalb der IT-Infrastruktur bereits vorhandene Komponenten für die Benutzerverwaltung, das Dokumentenmanagement und die Datensicherung sowie andererseits die innerhalb des RAfEG-Softwaresystems integrierten Komponenten zur Workflowsteuerung, Kommunikation und Präsentation. Die genannten Komponenten sind dabei austauschbar, sofern diese die innerhalb des RAfEG-Softwaresystems definierten,

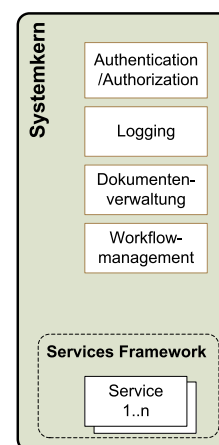


Abbildung 6: Systemkern

offenen Schnittstellen unterstützen. Das RAfEG-Softwaresystem ist damit auf beliebige Institutionen portierbar, sofern diese die notwendigen Schnittstellen bieten.

**Präsentationskomponente:** Die Funktionalität des RAfEG-Softwaresystems wird dem Nutzer durch eine Präsentationskomponente zur Verfügung gestellt.

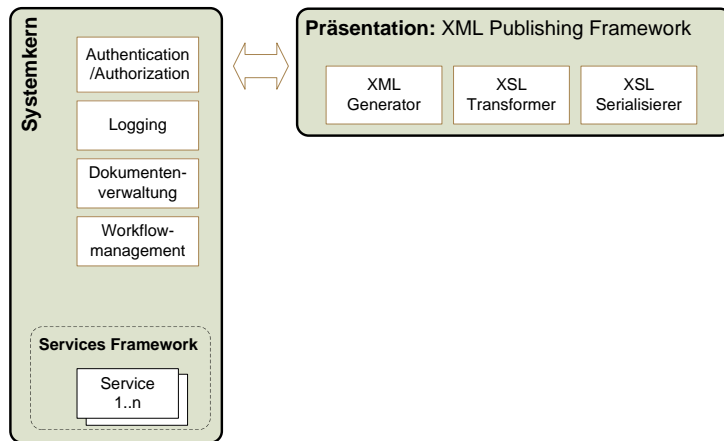


Abbildung 7: Präsentationskomponente des Baukastensystems

innerhalb der komponentenbasierten Architektur in die Präsentationsschicht integriert. Das RAfEG-Softwaresystem setzt dabei auf das von der Apache Software Foundation implementierte Cocoon Framework [Nie06]. Cocoon stellt einen Ansatz zur Verfügung, der auf Basis von XML-Dateien in mehreren Schritten die eigentlichen Daten in verschiedene Ausgabeformate transformiert. Als Eingabe dient dabei eine XML-Datei, die eine reine Repräsentation der Daten ohne Layoutinformationen enthält. Nachdem diese Daten in einem ersten Schritt eingelesen wurden, werden sie im nächsten Schritt um Layoutinformationen erweitert. Die Layoutinformationen sind dabei allerdings noch nicht an ein bestimmtes Ausgabeformat gebunden. Das Hinzufügen von Layoutinformationen bedeutet hierbei z. B. die Aufbereitung von Informationen in Form von Tabellen oder das Hinzufügen von Überschriften und Informationstexten. Die Zwischendarstellung der Daten mit Informationen zum Layout wird in einem letzten Schritt in das eigentliche Ausgabeformat umgewandelt, z. B. X/HTML oder PDF. Neben der Umwandlung von Daten in ein anzeigbares Dokument unterstützt Apache Cocoon weiterhin die Definition mehrerer Sprachdateien, um die Ausgabe in einer vom Nutzer gewünschten Sprache anzeigen zu können ohne die Layoutinformationen anpassen zu müssen. Definiert wurden innerhalb des RAfEG-Softwaresystems bisher die Ausgabe in Deutsch und Englisch.

**Kommunikationskomponente:** Zur Unterstützung der Interaktion verschiedener an der Abarbeitung eines Workflows beteiligter Institutionen sind Kommunikationsdienste notwendig. Herkömmliche Architekturen unterstützen dabei in der Regel lediglich einfache elektronische Dienste (E-Mail, Instant Messaging, etc.). Im Hinblick auf die Heterogenität der IT-Infrastruktur und die Anforderungen behördlicher Verwaltungsverfahren ist jedoch eine flexible Wahl des Kommunikationsweges zu gewährleisten, welche die technischen (Bandbreite, synchrone/asynchrone Kommunikation) und organisatorischen (gesetzliche Vorgaben etc.) Gegebenheiten berücksichtigt. Die Kommunikationskomponente stellt innerhalb des RAfEG-Softwaresystems die Interaktion zwischen den

am Workflow beteiligten Institutionen sicher. Während der Ausführung einer Aufgabe wird von der Workflowsteuerungssystemkomponente die Anforderung einer Zuarbeit an die Kommunikationskomponente übergeben. Aus der hinterlegten Konfiguration wird der zu wählende Kommunikationsweg bestimmt. Hierfür kommen verschiedene Technologien (z.B. WebServices, RMI, CORBA) in Frage, die sich auch kombinieren lassen. Auch herkömmliche Kommunikationsformen wie E-Mail, Fax, Post oder Kurier müssen unterstützt werden. Eine Unterstützung von E-Mail-Kommunikation ist realisiert. Die Möglichkeiten für Post und Kurier müssen weiterhin manuell erledigt werden, da für diese Kommunikationsarten für ein elektronisches System keine Möglichkeiten vorhanden sind. Die Unterstützung einer Vielzahl von Kommunikationsformen stellt einen effizienten Austausch von Dokumenten und Informationen sicher.

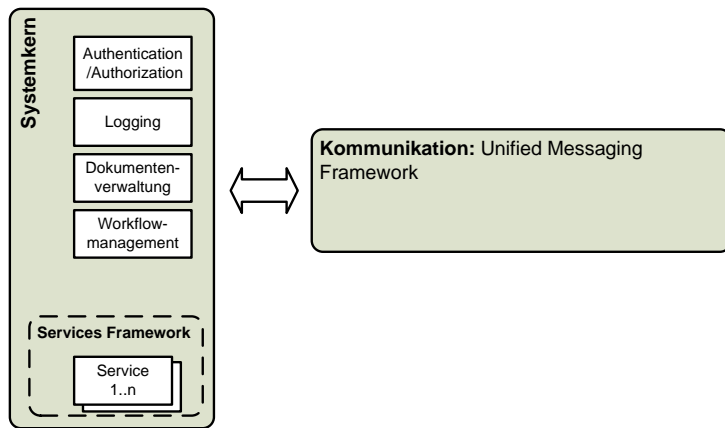


Abbildung 8: Kommunikationskomponente des Baukastensystems

Kurier müssen weiterhin manuell erledigt werden, da für diese Kommunikationsarten für ein elektronisches System keine Möglichkeiten vorhanden sind. Die Unterstützung einer Vielzahl von Kommunikationsformen stellt einen effizienten Austausch von Dokumenten und Informationen sicher.

**Workflowsteuerungssystemkomponente:** Die Workflowsteuerungssystemkomponente umfasst alle Funktionen, die mit der Abarbeitung von als Workflows modellierten Verwaltungsvorgängen zusammenhängen. Diese Komponente arbeitet dabei

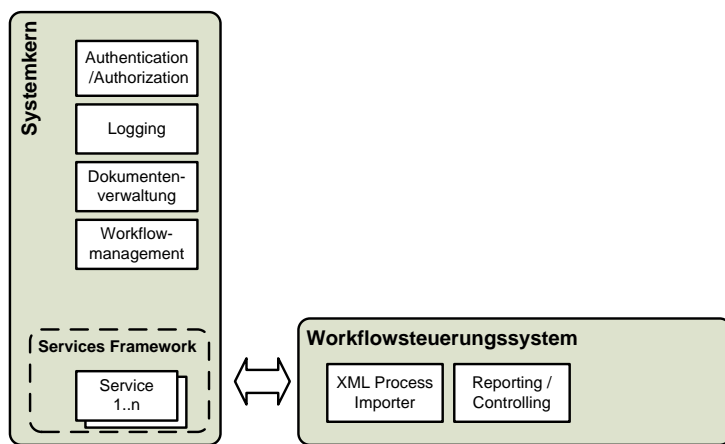


Abbildung 9: Workflowsteuerungssystemkomponente des Baukastensystems

direkt mit dem WfMS zusammen. Im Falle des Prototyps sind derzeit zwei WfMS nutzbar, Enhydra Shark [Enh05] und WfMOpen [LSW13]. Enhydra Shark und auch WfMOpen werden dabei von dieser Komponente als Bibliothek benutzt. Das bedeutet, dass das eingesetzte Workflowmanagementsystem immer in dieser Komponente gekapselt ausgeführt wird und die Komponente von einem konkreten WfMS unabhängige Zugriffsmethoden auf die Funktionalität des WfMS bietet. Somit ist ein Austausch des WfMS leicht möglich. Die genauen Aufgaben der Komponente sind:

- Laden von Prozessdefinitionen über die Schnittstelle für das Modellierungstool,
- Anlegen von Abbildungsvorschriften von in der Prozessdefinition angegebenen generischen Dokumentennamen auf existierende Dokumente,
- Starten von Prozessen,



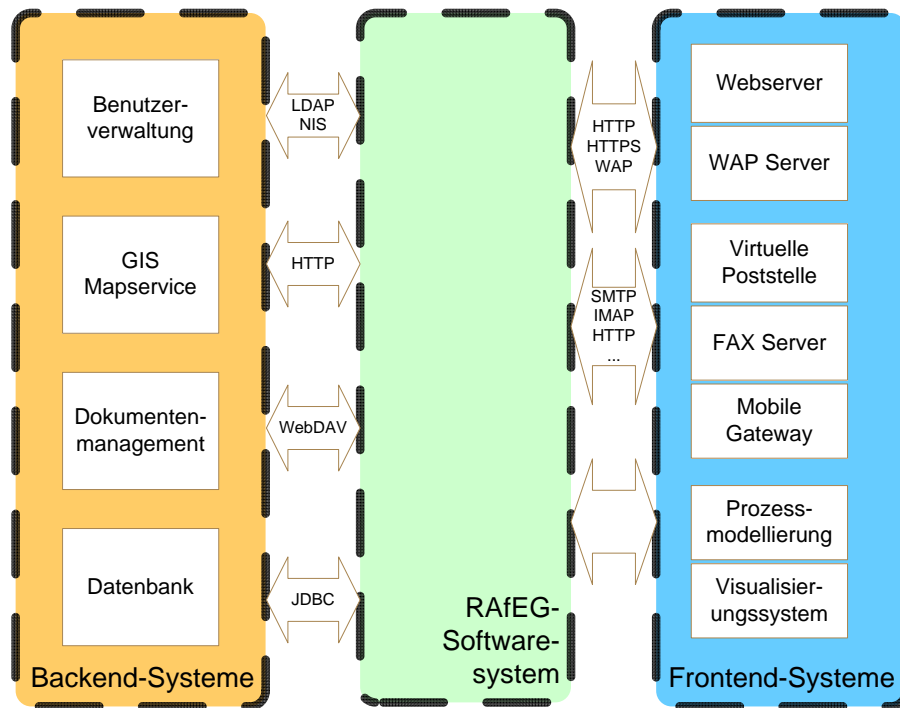


Abbildung 10: Frontend- und Backend-Systeme des RAfEG- Prototypen.

- Anlegen der notwendigen Informationen für Aktennotizen zu einem Prozess,
- Bereitstellen einer Übersicht aller zu einem bestimmten Zeitpunkt ausgeführten Prozesse,
- Anlegen von Abbildungen realer Systemnutzer auf in der Prozessbeschreibung definierte Nutzer/Rollen,
- Anzeigen von Aufgabenlisten für die einzelnen Nutzer,
- Bereitstellung von Möglichkeiten der Interaktion von Nutzern mit den Aufgaben (Annahme, Aktualisierung, Beendigung),
- Lesen und Schreiben von Dokumenten einer Aufgabe durch einen Nutzer und
- Bereitstellung von Funktionen für eine automatisiert ablaufende verteilte Abarbeitung von Prozessen.

Zum Abschluss der Beschreibung des Baukastensystems werden jetzt das Frontend- und das Backend-System vorgestellt. Abbildung 10 zeigt diese mit den Protokollen, die zur Anbindung genutzt wurden.

**Backend-Systeme (Nachgeschaltete Komponenten):** Als nachgeschaltete Komponenten werden die Komponenten bezeichnet, die zwar grundlegende Funktionen für das RAfEG-Softwaresystem bereitstellen und zwingend notwendig, jedoch in bestehenden IT-Infrastrukturen meist bereits vorhanden sind. Dazu gehören die Benutzerverwaltung, die zugrunde liegende Datenbank, das geographische Informationssystem und das Dokumentenmanagementsystem. Die Referenzimplementierung stellt Beispielimplementierungen dieser Komponenten bereit und nutzt offene Schnittstellen und Standards zur Anbindung (z.B. LDAP und NIS für die Benutzerverwaltung, JDBC zur Datenbankanbindung), was einen einfachen Austausch bei der Einführung des RAfEG-Softwaresystems in Behörden ermöglicht.

*Dokumentenmanagement:* Die Verwaltung der elektronischen Dokumente wird von einem Dokumentenmanagementsystem übernommen, welches Funktionen zur Archivierung, Versionierung und Änderungsverfolgung sowie Möglichkeiten zur Vergabe von Berechtigungen auf Dokumente bereitstellt. Für die Erstellung/Erfassung, Änderung, Prüfung, Verteilung, Verwendung und Archivierung von Dokumenten wird das DMS über eine WebDAV (Web-based Distributed Authoring and Versioning)-Schnittstelle angesprochen [GWF<sup>+</sup>99, CRSW04]. Im Vordergrund der WebDAV-Nutzung steht, dass sich der Einsatz dieses offenen Standards positiv auf die geforderten Eigenschaften der Interoperabilität und Portabilität innerhalb der vorliegenden heterogenen Systemlandschaft auswirkt. Der Hauptgrund dafür ist, dass zahlreiche kommerzielle und Open Source Produkte WebDAV unterstützen, so dass vorhandene Systeme eingebunden bzw. kostengünstige Lösungen eingeführt werden können. Über integrierte Erweiterungen von WebDAV wird Konfigurationsmanagement/Versionierung und Dokumentensuche ermöglicht. Das Dokumentenmanagementsystem wird an erster Stelle für die Verwaltung der Prozessmappen der einzelnen Verwaltungsvorgänge benötigt. Das Konzept der Prozessmappen wird in [KRS07b, KRS07a] detailliert vorgestellt. In einer Prozessmappe sind alle Dokumente eines gestarteten Verwaltungsvorgangs abgelegt. Detaillierte Informationen zur Dokumentenverwaltung sind in [KRS06] zu finden.

*Benutzerverwaltung:* Zur Senkung des Administrationsaufwandes bei der Verwaltung der Benutzer ist die Kopplung des RAfEG-Softwaresystems mit der zentralen Benutzerverwaltung der jeweiligen Institution vorgesehen. Die Anbindung erfolgt über das Lightweight Directory Access Protocol (LDAP) [WHK97], welches von einer Vielzahl von Verzeichnisdiensten unterstützt wird. Andere primär verwaltungsintern genutzte Verzeichnisdienste (ActiveDirectory, NDS etc.) bieten i.d.R. einen LDAP Zugriff, so dass bestehende verwaltungsinterne Verzeichnisdienste problemlos direkt angebunden werden können. Mit dem Ziel einer hohen Nutzerakzeptanz vereinfacht dies nicht nur die Administration, sondern erleichtert gleichzeitig den Nutzern den Zugriff auf das System, da die Zugangsdaten des Benutzers ebenfalls für das RAfEG-Softwaresystem gelten.

*Geographisches Informationssystem:* Wie in Abschnitt 2.1 bereits erwähnt, gibt es eine Vielzahl von geographischen Informationssystemen, die zumeist ihre eigenen Formate zur Datenspeicherung besitzen. Der Lösungsansatz innerhalb des RAfEG-Backend-Systems ist daher ein Format zu nutzen, in das von den einzelnen Formaten konvertiert werden kann und für dieses einen eigenen Server, der die Daten für den RAfEG-Systemkern bereitstellt, zu nutzen. Als Format hat sich dabei das ShapeFormat [Env98] angeboten, da es sich zu einem De facto-Standard im Bereich von GIS entwickelt hat. Die Shapefiles werden dabei auf einem freien Web Map Server gespeichert. Bei der Anfrage einer bestimmten Karte an diesen werden die Daten zuerst in ein anzeigbares Format konvertiert und dann über die Präsentationskomponente an den anfordernden Nutzer gesendet. Als freier Web Map Server wird dabei deegree [DEE] eingesetzt.

*Datenbank:* Die Speicherung systeminterner Daten erfolgt in einer relationalen Datenbank. Die Anbindung an das System erfolgt über JDBC (Java Database Connectivity) [Sun], einer einheitlichen Schnittstelle zu frei verfügbaren bzw. kommerziellen Datenbanksystemen verschiedener Hersteller. In der derzeitigen Implementierung wird das freie Datenbankmanagementsystem Postgres [Mom01] verwendet.

**Frontend-Systeme (Schnittstellen nach Außen):** Als Schnittstellen nach Außen werden solche Komponenten bezeichnet, die eine Interaktion mit dem RAfEG-Softwaresystem

gestatten. Die wichtigste ist dabei die auf X/HTTP und WAP basierende Nutzerschnittstelle, über die Nutzer mit dem System arbeiten können und das System administriert werden kann. Weiterhin ist eine Schnittstelle zur Kommunikation zwischen RAfEG-Systemen in verschiedenen Behörden bzw. zwischen Behörden und externen Institutionen implementiert, die eine verteilte Abarbeitung von Verwaltungsvorgängen gestattet. Als letzte Komponente der externen Schnittstellen ist die für den Import von Prozessdefinitionen (als Workflow modellierte Verwaltungsvorgänge) verwendete zu nennen, die es ermöglicht Workflows in das System zu integrieren.

*Nutzerschnittstelle:* Das innerhalb des RAfEG-Systems vorauszusetzende heterogene IT-Systemumfeld legt den Einsatz eines plattformunabhängig nutzbaren Benutzerinterfaces nahe. Durch die gleichzeitig fast durchgängige Verfügbarkeit von Webbrowsern und verwaltungsinternen Netzwerken bzw. der Anbindung externer Institutionen an das Internet ist die Nutzung eines webbasierten Interfaces zu empfehlen. In diesem Bereich scheiden alternative Technologien (bspw. Java Swing, X11) auf Grund der eingesetzten IT-Infrastruktur aus. Der Anwendungsserver muss daher in der Lage sein, eine Oberfläche in einem der gängigen Ausgabeformate zu generieren sowie Eingaben des Benutzers an die Komponenten einer Applikation zu übermitteln, welche für die Bearbeitung der Eingaben zuständig sind. Mit dem Ziel der Wiederverwendbarkeit und Kapselung wird damit jede Funktion als eigenständige Komponente innerhalb eines Frameworks abgebildet. In der Bundesrepublik Deutschland regeln die Barrierefreie Informationstechnik-Verordnung (BITV) [Bun02b] und der §7 Behindertengleichstellungsgesetz (BGG) [Bun02a] die wesentlichen Kriterien zur Erstellung webbasierter Benutzeroberflächen in öffentlichen Verwaltungen. Diese Richtlinien laufen letztlich auf die WAI (Web Accessibility Initiative)-Richtlinien [W3C02] hinaus. In Bezug auf die Ergonomie sind in erster Linie die Regelungen des Bundesarbeitsschutzgesetzes und die Bildschirmarbeitsverordnung bindend. Nach §47 Bildschirmarbeitsverordnung (Umsetzung der EU-Richtlinie 90/270 EWG) müssen Oberflächen, welche zur Erledigung „fremdbestimmter Aufgaben“ zur Erreichung von Zielen einer Verwaltung bestimmt sind, Angaben zu Dialogabläufen und die Beschreibung von Fehlern anbieten. Neben den Gesetzen befassen sich mehrere internationale, europäische und deutsche Normen mit der Ergonomie. Zu den wichtigsten Normen gehören:

- DIN 29241: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten und
- DIN 66234, Teil 8 und ISO 9241, Part 10: Suitability for the task, Self-descriptiveness, Controllability, Conformity with user expectations, Error tolerance, Suitability for individualization, Suitability for learning.

Beim Aufbau der Benutzerschnittstelle des RAfEG-Softwaresystems wurden diese Normen berücksichtigt.

*Prozessmodellierung und Import von modellierten Verwaltungsprozessen:* Die Verwaltungsvorgänge, im speziellen das Planfeststellungsverfahren, wurden vom Institut für Wirtschaftsinformatik (IWi) Saarbrücken in enger Zusammenarbeit mit dem Regierungspräsidium Leipzig erstellt. Als Modellierungstool wurde hierzu ARIS verwendet. Das primäre Format der Prozessdefinition ist damit AML (Aris Modelling Language). Da AML jedoch nicht direkt als Eingabe eines Open Source WfMSs genutzt werden kann, wurde durch die Professur Praktische Informatik der TU-Chemnitz eine Umwandlung von AML nach XPD L durchgeführt, da die eingesetzten Workflowmanagementsysteme (Enhydra Shark und WfMOpen) die von der WfMC spezifizierte Sprache XPD L [Nor02] als Eingabe von Prozessdefinitionen nutzen. AML ist jedoch ein proprietäres Format, weshalb

ein Zwischenschritt über EPML (Event Driven Process Chain Markup Language) [MN05] vollzogen wurde, da ein Tool für die Konvertierung dieser beiden Formate frei verfügbar ist. Die Konvertierung von EPML nach XPDL konnte durchgeführt werden, da EPML ein offener Standard und gut dokumentiert ist. Eine genaue Darstellung der Umwandlung von AML nach XPDL ist in [BDR06, BDR07] zu finden. Die Einbindung von Prozessdefinitionen erfolgt über die Administrationsschnittstelle des WfMS. Dazu wird zuerst die XPDL-Beschreibung an das System gesendet. Ist die Beschreibung dem System dadurch zugänglich, kann sie im Falle einer neuen Beschreibung geladen und im Falle eines Updates einer bestehenden Beschreibung aktualisiert werden.

*Kommunikation mit anderen Systemen:* Die Kommunikation mit anderen Systemen wird im RAfEG-Softwaresystem über das OSCI-Transport Protokoll ermöglicht. Kommunikation mit anderen Systemen/Institutionen ist dabei notwendig, wenn *a)* die Komponenten eines RAfEG-Softwaresystems über mehrere Server verteilt werden, *b)* Workflows verteilt abgearbeitet werden und *c)* Dokumente oder Karten des geographischen Informationssystems an externe Institutionen versendet werden müssen. OSCI-Transport bietet dabei zum einen die notwendigen Sicherheitsmechanismen, die im Bereich der öffentlichen Verwaltungen gefordert werden, und zum anderen ermöglicht es eine transparente und nachvollziehbare Protokollierung der Kommunikation. Zusätzlich ist es unabhängig vom eigentlichen Transportprotokoll und ermöglicht so die Kommunikation mit verschiedensten Systemen. Dies ist wichtig, da die verschiedenen Behörden und externen Institutionen in den meisten Fällen nicht die gleichen technischen Voraussetzungen für Kommunikationsprotokolle haben. So können z. B. einzelne Behörden nur Kommunikation über E-Mail zulassen, wogegen andere z. B. nur SOAP (Simple Object Access Protocol) [W3C07b] als Protokoll zum Datenaustausch bereitstellen.

## 4.4 Sicherheitsaspekte

Sicherheit ist für Government-to-Government (G2G) Anwendungen von zentraler Bedeutung. Deshalb wurde bei der Konzeption und prototypischen Realisierung auf diesen Punkt besonders eingegangen. Im Wesentlichen sind hier zwei Hauptbereiche, in denen Sicherheitsmechanismen zum Einsatz kommen, zu erwähnen. Der erste Punkt ist die Sicherheit, die bei der Abarbeitung von Verwaltungsvorgängen innerhalb einer Behörde vorausgesetzt werden muss. Dies umfasst die Punkte:

- Nutzerverwaltung mit der Möglichkeit der Vergabe von Restriktionen zwischen verschiedenen Benutzergruppen und
- Absicherung einzelner Komponenten des Baukastensystems des RAfEG-Softwaresystems, sofern diese verteilt über mehrere Server der Behörde agieren.

Die Nutzerverwaltung wurde über das in Abbildung 11 dargestellte Authentifizierungssystem des RAfEG-Softwaresystems realisiert. Ein Nutzer, Mitarbeiter der Behörde, meldet sich über einen Browser beim RAfEG-Softwaresystem an. Dabei werden über den Login-Context der Präsentationskomponente die Anmeldedaten an den RAfEG-Systemkern – genauer die Authentifizierungsschnittstelle – übergeben. Innerhalb des Systemkerns werden die eingegebenen Daten gegen die in der Nutzerverwaltung hinterlegten geprüft. Als Nutzerverwaltung kann hier jedes System, das über eine LDAP-Schnittstelle verfügt, angesprochen werden. Das erlaubt den Einsatz des RAfEG-Softwaresystems in bestehenden Systemlösungen.

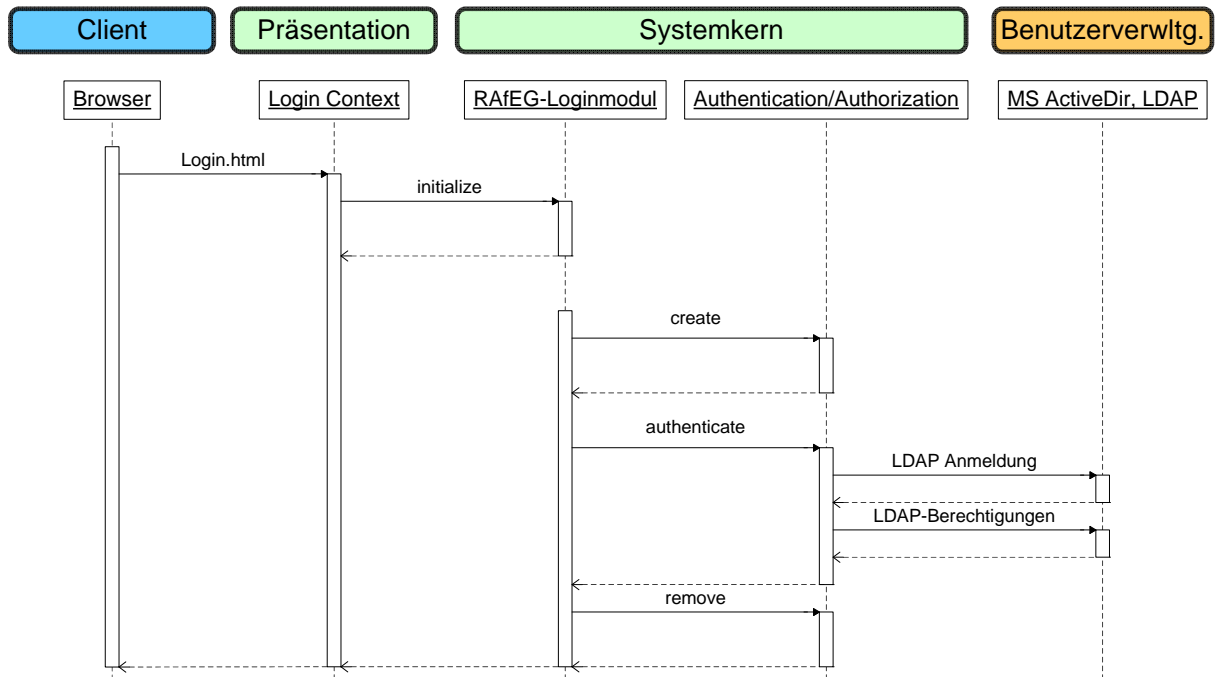


Abbildung 11: Authentifizierungssystem für Nutzer des RAfEG-Softwaresystems.

Sicherheit zwischen einzelnen, auf verschiedenen Servern ausgeführten Komponenten des RAfEG-Softwaresystems einer Behörde – genutzt für Lastverteilung oder Absicherung einzelner Komponenten – wird über den verwendeten Applikationsserver und die Enterprise JavaBean-Technologie bereitgestellt. Im Falle der prototypischen Implementierung ist dies der JBoss Application Server (JBoss AS). Durch diese Konstellation ist es möglich, die Methoden der Schnittstellen mit fest definierten Rollen zu koppeln und somit den Nutzern nur den Zugriff auf Funktionen zu gestatten, die zu ihrer Rolle/ihren Rollen passen. Dabei ist als Grundlage zu erwähnen, dass das System und die Kommunikation zwischen Systemteilen auf Nutzeraktionen beruht. Das bedeutet, dass die einzelnen Teile des Baukastensystems nur miteinander kommunizieren müssen, wenn ein Nutzer eine Anfrage an das System stellt. Dadurch können die einzelnen Komponenten untereinander abgesichert werden, da nur authentifizierte Benutzer eine Kommunikation einleiten dürfen. Im Falle des Nichtbesitzens einer Rolle für eine bestimmte Schnittstellenmethode, wird der Zugriff verweigert.

Der zweite Punkt, bei dem die Sicherheit eine Rolle spielt ist das externe Abarbeiten von Teilprozessen. Hier kommt das in Punkt 3.1 vorgestellte Client/Servermodell zum Einsatz. Die Kommunikation erfolgt in diesem Fall über OSCI-Transport. Dies erlaubt sowohl eine sichere, als auch nachvollziehbare Kommunikation zwischen den Kommunikationskomponenten zweier RAfEG-Softwaresysteme. Nachvollziehbare Kommunikation ist dabei für Software in öffentlichen Verwaltungen von hoher Priorität, um im Nachhinein die Interaktion zwischen Behörden verfolgen zu können. In der erstellten prototypischen Realisierung kommt die Kommunikationsvariante “One-WayMessage, passiver Empfänger” zum Einsatz. In dieser Variante überträgt der Sender seine Nachricht an einen Intermediär. Dieser wird innerhalb von OSCI-Transport vorgeschrieben, da keine direkte Verbindung zwischen zwei Systemen stattfinden darf. Der Intermediär leitet die Nachricht an den Empfänger weiter und protokolliert den Empfang und das Weitersenden (Nachvollziehbarkeit). Der Empfänger muss einen so genannten *AcceptDeliveryHandler* implementieren,

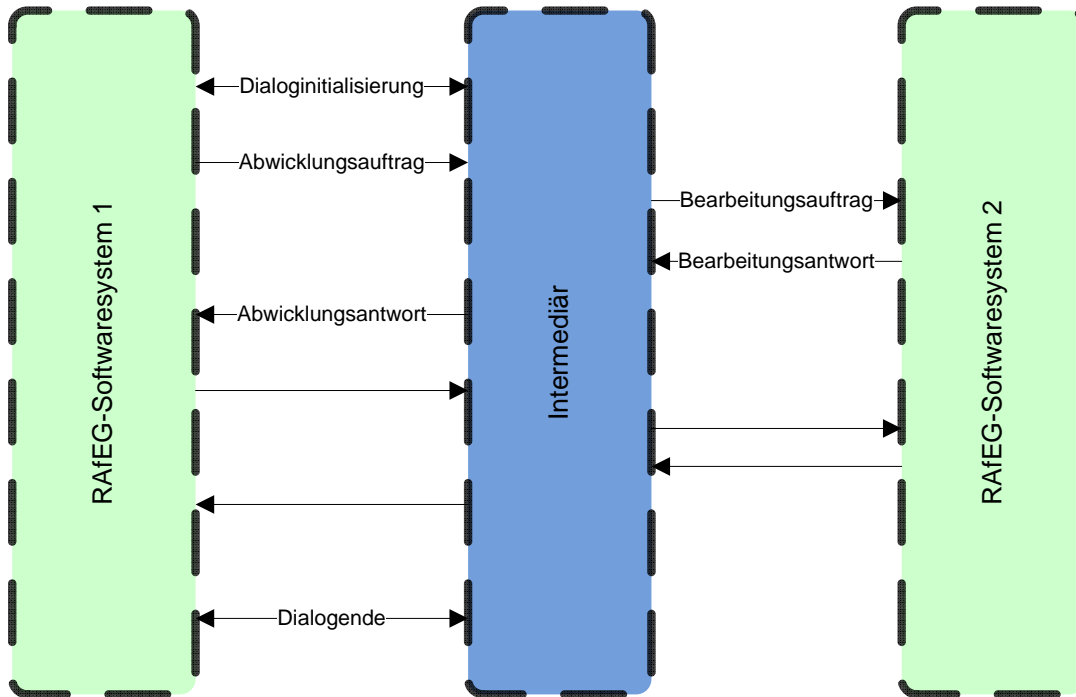


Abbildung 12: Kommunikationsablauf bei Nutzung von OSCI-Transport

der eine solche Nachricht annimmt, an die Applikation weitergibt und den Empfang bestätigt (quittiert). Die Quittung erhält wieder der Intermediär und leidet sie an den Sender weiter. Ein schematischer Überblick der Kommunikation zwischen zwei RAfEG-Systemen über OSCI-Transport ist in Abbildung 12 zu sehen.

OSCI-Transport schreibt kein festes Kommunikationsprotokoll vor und ermöglicht somit den Einsatz beliebiger Transportprotokolle. Für die Ver- und Entschlüsselung der übermittelten Daten kommen im RAfEG-Softwaresystem verschiedene Verfahren zum Einsatz. Das Besondere an der Herangehensweise innerhalb des RAfEG-Softwaresystems ist, die zu übermittelnden Daten in Sicherheitsstufen einzuteilen, was zu optimaler Effizienz – große Daten mit niedriger Sicherheitsstufe müssen nicht aufwendig verschlüsselt werden – und maximaler Sicherheit – Einsatz starker Kryptoverfahren für Daten mit hohen Sicherheitsanforderungen – führt. Je höher dabei die Sicherheitsanforderung an die Daten (z.B. Dokumente und Formulare) ist, desto stärker ist der genutzte Verschlüsselungsalgorithmus.

## 4.5 Externe Datenrepräsentation

Das Grundkonzept der externen Datenrepräsentation sieht eine strikte Trennung zwischen Anwendungs- und Kommunikationslogik vor. Das bedeutet, dass keine Informationen über die sendende und die empfangende Anwendung in die versendeten Daten einfließen. Entsprechend der Anforderungen an das RAfEG-Softwaresystem wurden eine XML-basierte Repräsentation und die Übermittlung durch SOAP [W3C07b] gewählt. Eine Übermittlung von Daten mit Möglichkeiten wie CORBA [Obj04] bzw. Java Remote Method Invocation (RMI) [Gro01] hätte einen einschränkenden Einfluss auf die externe Datenrepräsentation. Bei der Nutzung von RMI ist man an das Versenden von Java-Objekten gebunden, was den Einsatz eines Systems in einer anderen Programmiersprache unmöglich macht und die

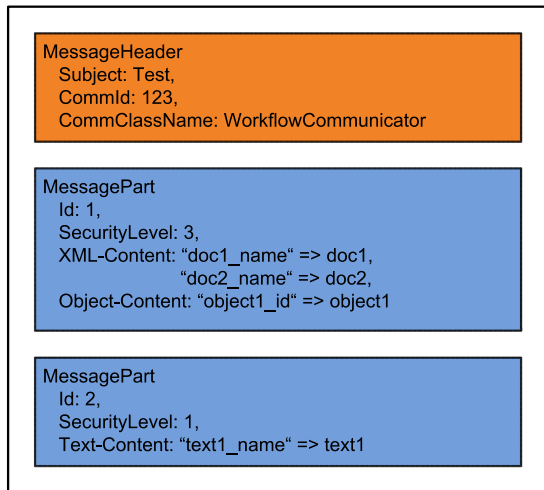


Abbildung 13: Interne Struktur einer Nachricht

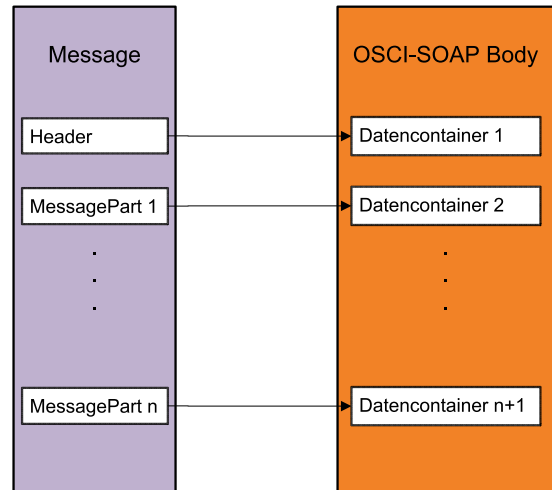


Abbildung 14: Verpackung der zu versendenden Daten in die OSCI - SOAP Nachricht

Anbindung beliebiger Systeme zu stark einschränkt, da in diesem Fall beide Kommunikationspartner in Java implementiert werden müssen. CORBA hat den Nachteil, dass die Schnittstellen in der Regel mittels einer Interface Definition Language (IDL) beschrieben werden müssen und eine formale Spezifikation der Klassen und Objekte sowie sämtlicher Parameter und Datentypen erzeugt wird. Diese Schnittstellenbeschreibung wird dann in ein Objektmodell der verwendeten Programmiersprache umgesetzt. Hier müssen zu viele Informationen über die kommunizierenden Anwendungen mit einfließen. Ein weiterer Nachteil von CORBA ist, dass es eine Vielzahl von Implementierungen verschiedener Hersteller gibt, die oftmals nicht zueinander kompatibel sind.

Aus den aufgeführten Gründen wurde eine XML-basierte externe Datenrepräsentation gewählt, da XML-Verarbeitung von den gebräuchlichsten Programmiersprachen unterstützt wird. Dabei können durch die erstellte Schnittstelle jedoch weiterhin gekapselte serialisierte Java-Objekte versendet werden, wenn eine Anwendung dies fordert. Der durchgeführte Ansatz mit dem internen RAfEG-spezifischen und dem von OSCI-Transport versendeten Nachrichtenformat sind in Abbildung 13 und Abbildung 14 zu sehen.

Durch den Einsatz von OSCI-Transport als Kommunikationsprotokoll, und der damit einhergehenden Nachrichtenversendung von XML-Nachrichten über SOAP lag es nahe, für die interne Repräsentation der Daten auch XML zu verwenden. Eine Nachricht besteht dabei mindestens aus einem Kopf (*MessageHeader*) bestehend aus einem Betreff (*Subject*), dem Kommunikatorklassenname (*CommClassName*) und einem Kommunikationsidentifikator (*CommId*). Das Subject-Feld beinhaltet einen Betreff der Nachricht, der von der Applikation frei gewählt werden kann. Der Kommunikationsidentifikator entspricht dem Identifikator des übergeordneten Kommunikationsvorgangs, zu dem die aktuelle Übertragung gehört. Der Kommunikatorklassenname wird benötigt, um den richtigen Kommunikator auf der Empfängerseite nutzen zu können. Prinzipiell ist es möglich, beliebige Kommunikatoren selbst zu implementieren, die zu der im RAfEG-Softwaresystem definierten Kommunikatorschnittstelle passen. Neben dem Nachrichtenkopf können beliebig viele Datenteile (*MessageParts*) in die zu versendende Nachricht eingebaut werden. Jeder Nachrichtenteil besteht aus einem eindeutigen Identifikator (*Id*) und einem Sicherheitslevel (*SecurityLevel*), der zwischen 1 (niedrige Sicherheitsstufe) und 3 (höchste Sicherheitsstufe) liegen kann. Informationen zu Sicherheitsleveln und ihrer Anwendung werden

in [KRS07a, KRS07b] vorgestellt. Zusätzlich werden noch die Daten in den Datencontainer gepackt, was Textdaten (*Text-Content*), Strukturierte Daten (*XML-Content*) oder Dateien und sonstige Binärdaten (*Object-Content*) sein können. Durch die Möglichkeit des Mischens von Daten in den Datenteilen kann bei der nutzenden Implementierung relativ frei gewählt werden, wie die Daten verschickt werden sollen (z. B. jeder Typ in einen einzelnen Datenteil oder gemischte Datenteile aus Text und strukturierten Daten). Dadurch können die einzelnen Datenteile an eigene Bedürfnisse angepasst werden. Die Abbildung der Daten des internen Formats auf die Datencontainer des OSCI-Transports erfolgt in der Art und Weise, dass jeder Nachrichtenteil (*Messagepart*) auf einen Datencontainer abgebildet wird. Das bietet neben dem einfachen Verständnis der tatsächlich versendeten Daten zusätzlich die Möglichkeit einzelne Nachrichtenteile getrennt voneinander zu sichern, indem man die Sicherheitsmechanismen pro Datencontainer einsetzt. Detaillierte Informationen zu den Datenformaten und der Übertragung sind in [Jun06] zu finden.

## 4.6 Anpassung/Evaluierung

Eine Anpassung/Evaluation des Prototyps des RAfEG-Softwaresystems erfolgte während der Implementierung an der Technischen Universität Chemnitz. Durch den parallelen Ansatz der gleichzeitigen Implementierung und Testphasen der einzelnen Bausteine konnte hier frühzeitig die Interoperabilität und Praxistauglichkeit der Schnittstellen überprüft werden. Die prozessbezogene Adaption des Prototyps erfolgte in Abstimmung mit dem Institut für Wirtschaftswissenschaften Saarbrücken. Hier waren insbesondere Anpassungen der in ARIS modellierten Prozesse für die Konvertierung in XPDL notwendig. Die vorgesehene Evaluierung durch das Regierungspräsidium Leipzig steht noch aus. Der Prototyp des RAfEG-Softwaresystems wurde jedoch am Regierungspräsidium Leipzig installiert und den Mitarbeitern in mehreren Präsentationen vorgestellt.

## 5 Wissenstransfer

Die an der Professur Praktische Informatik der Technischen Universität Chemnitz entstandenen Arbeiten im Bereich der Konzeption, Analyse und Implementierung eines komplexen Workflowmanagementsystems zur verteilten Abarbeitung langlebiger Prozesse im E-Government wurde in mehreren Konferenzbeiträgen [BHP<sup>+</sup>05, BKR05, BKR06, BDR07, KRS07a, KRS07b] und Berichten [BHRV05, BHK<sup>+</sup>06, KRS06, BDR06] dargestellt.

Durch das adaptive Konzept des RAfEG-Softwaresystems ist es möglich, neben Verwaltungsvorgängen im Bereich des E-Government, allgemeine langlebige Prozesse verteilt abzuarbeiten. Dadurch wird der Einsatz sowohl des entstandenen Konzeptes, als auch des entstandenen Softwaresystems für Projekte außerhalb des E-Government interessant. Der ausschließliche Einsatz von Open Source Software mit offenen Schnittstellen und offener Protokolle erlaubt es zudem, das implementierte Softwaresystem in bestehende Systemumgebungen zu integrieren und an spezielle Bedürfnisse anzupassen. Die Forschungskompetenz der Professur Praktische Informatik wurde im zukunftssträchtigen Bereich der parallelen und verteilten Softwareentwicklung ausgebaut.



## 6 Projektbeteiligte

Projektbeteiligte an der Technischen Universität Chemnitz waren Daniel Beer, Jörg Dümmler, Steffen Höhne, Raphael Kunis, Gudula Rüniger, Michael Schwind und Michael Voigt.

Konsortiumspartner innerhalb des BMBF Projektes waren das Institut für Wirtschaftsinformatik (IWi) im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI), die NHConsult GmbH und das Regierungspräsidium Leipzig.

## Literatur

- [Ado04] ADOBE SYSTEMS INCORPORATED: *PDF Reference Fifth Edition – Adobe Portable Document Format, Version 1.6*. <http://partners.adobe.com/public/developer/en/pdf/PDFReference16.pdf>. Version: 2004
- [Alk07] ALKACON SOFTWARE GMBH: *OpenCms*. <http://www.opencms.org/opencms/>. Version: 2007
- [Apa] APACHE JAKARTA PROJECT: *Jakarta Slide*. <http://jakarta.apache.org/slide/architecture.html>
- [BDR06] BEER, D. ; DÜMMLER, J. ; RÜNGER, G.: Automatische Erzeugung von Workflowbeschreibungen im XPDL-Format aus ereignisgesteuerten Prozeßketten / Technische Universität, Fakultät für Informatik. 2006. – Chemnitzer Informatik-Berichte CSR-06-11
- [BDR07] BEER, D. ; DÜMMLER, J. ; RÜNGER, G.: Transformation von Prozessmodellen in Workflowbeschreibungen. In: GEIST (Hrsg.) ; HEINDL (Hrsg.) ; SCHWEIGHOFER (Hrsg.): *Tagungsband des 10. Internationalen Rechtsinformatik Symposions IRIS 2007*, BOORBERG, 2007
- [BHK<sup>+</sup>06] BEER, D. ; HÖHNE, S. ; KUNIS, R. ; RÜNGER, G. ; VOIGT, M.: RAFEG - Eine Open Source basierte Architektur für die Abarbeitung von Verwaltungsprozessen im E-Government / Technische Universität, Fakultät für Informatik. 2006. – Chemnitzer Informatik Berichte CSR-06-03
- [BHP<sup>+</sup>05] BEER, D. ; HÖHNE, S. ; PETERSOHN, H. ; PÖNITZSCH, T. ; RÜNGER, G. ; VOIGT, M.: Designing a Distributed Workflow System for E-Government. In: *Proceedings of the 24th IASTED International Conference on Modelling, Identification and Control*. Innsbruck, Austria, 2005
- [BHRV05] BEER, D. ; HÖHNE, S. ; RÜNGER, G. ; VOIGT, M.: Software- und Kriterienkatalog zu RAFEG - Referenzarchitektur für E-Government / Technische Universität, Fakultät für Informatik. 2005. – Chemnitzer Informatik-Berichte CSR-05-01
- [BKR05] BEER, D. ; KUNIS, R. ; RÜNGER, G.: An Interface for the Execution of Distributed Hierarchical Workflows. In: *Proc. of the 14th Euromicro Conference on Parallel, Distributed and Network based Processing (PDP'06)*. Montbéliard, France, 2005

- [BKR06] BEER, D. ; KUNIS, R. ; RÜNGER, G.: A Component Based Software Architecture for E-Government. In: *Proc. of the First International Conference on Availability, Reliability and Security (AReS 2006)*. Vienna, Austria, 2006
- [Bun01a] BUNDESMINISTERIUM DER JUSTIZ: *Gesetz über Rahmenbedingungen für elektronische Signaturen*. Version: Mai 2001. [http://www.gesetze-im-internet.de/sigg\\_2001/index.html](http://www.gesetze-im-internet.de/sigg_2001/index.html). –
- [Bun01b] BUNDESMINISTERIUM DER JUSTIZ: *Verordnung zur elektronischen Signatur*. [http://www.gesetze-im-internet.de/sigv\\_2001/index.html](http://www.gesetze-im-internet.de/sigv_2001/index.html). Version: November 2001
- [Bun02a] BUNDESMINISTERIUM FÜR GESUNDHEIT: *Gesetz zur Gleichstellung behinderter Menschen*. <http://bundesrecht.juris.de/bgg/index.html>. Version: Mai 2002
- [Bun02b] BUNDESMINISTERIUM FÜR GESUNDHEIT: *Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz*. <http://bundesrecht.juris.de/bitv/>. Version: Juli 2002
- [Bun03] BUNDESMINISTERIUM DES INNEREN: *SAGA 2.0: Standards und Architekturen für E-Government-Anwendungen*. KBSt-Schriftenreihe, Band 59. [http://www.kbst.bund.de/cln\\_012/nn\\_838538/SharedDocs/Publikationen-Kbst/KBSt-Publikationen/saga-version-2-0--kbst-schriftenreihe-band-59-.html](http://www.kbst.bund.de/cln_012/nn_838538/SharedDocs/Publikationen-Kbst/KBSt-Publikationen/saga-version-2-0--kbst-schriftenreihe-band-59-.html). Version: Dezember 2003
- [Bun05] *Kapitel II: Grundlagen - Verschlüsselung und Signatur*. In: BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (Hrsg.): *E-Government-Handbuch*. [http://www.bsi.bund.de/fachthem/egov/download/2\\_Krypto.pdf](http://www.bsi.bund.de/fachthem/egov/download/2_Krypto.pdf), 2005
- [CC 07] CC E-GOV GMBH: *CC DMS*. [http://www.cc-egov.de/cc\\_dms.htm](http://www.cc-egov.de/cc_dms.htm). Version: 2007
- [CRSW04] CLEMM, G. ; RESCHKE, J. ; SEDLAR, E. ; WHITEHEAD, J.: *RFC 3744*. <ftp://ftp.rfc-editor.org/in-notes/rfc3744.txt>. Version: 2004
- [DEE] *Deegree Web Map Server*. <http://deegree.sourceforge.net/index.html>
- [Enh05] ENHYDRA.ORG: *Shark 1.1 Documentation*. <http://shark.objectweb.org/doc/1.1/index.html>. Version: 2005
- [Env98] ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE, INC. (ESRI): *Shapefile Technical Description*. <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>. Version: Juli 1998
- [Gro01] GROSSO, William: *Java RMI*. O'Reilly Media, Inc., 2001
- [GWF+99] GOLAND, Y. ; WHITEHEAD, E. ; FAIZI, A. ; CARTER, S. ; JENSEN, D.: *RFC 2518: HTTP Extensions for Distributed Authoring – WEBDAV*. <ftp://ftp.rfc-editor.org/in-notes/rfc2518.txt>. Version: 1999. – Status: Proposed Standard

- [Hol95] HOLLINGSWORTH, D.: The Workflow Reference Model / Workflow Management Coalition. Version: Januar 1995. <http://www.wfmc.org/standards/docs/tc003v11.pdf> (Document Number: TC00-1003, Document Status – Issue 1.1). – The Workflow Management Coalition Specification. – Online-Ressource, Abruf: 02. 11. 2007
- [IDS] IDS SCHEER AG ; SCHEER, IDS (Hrsg.): *ARIS Markup Language (AML)*
- [Jam07] The Jam Warehouse Software Ltd.: *Knowledge Tree – User Manual*. 3.3.x. März 2007. [http://dfn.dl.sourceforge.net/sourceforge/kt-dms/KnowledgeTreeUserManual\\_2007-05-17.pdf](http://dfn.dl.sourceforge.net/sourceforge/kt-dms/KnowledgeTreeUserManual_2007-05-17.pdf)
- [JBo07] JBOSS: *JBoss Application Server*. <http://labs.jboss.com/jbossas/>. Version: 2007
- [Jun06] JUNGE, S.: *Sicherer Datenaustausch in E-Government-Anwendungen*, Technische Universität Chemnitz, Fakultät für Informatik, Diplomarbeit, 2006
- [KRS06] KUNIS, R. ; RÜNGER, G. ; SCHWIND, M.: Dokumentenmanagement für Verwaltungsvorgänge im E-Government / Technische Universität, Fakultät für Informatik. 2006. – Chemnitzer Informatik-Berichte CSR-06-10
- [KRS07a] KUNIS, R. ; RÜNGER, G. ; SCHWIND, M.: A new Model for Document Management in E-Government Systems Based on Hierarchical Process Folders. In: *Proc. of the 7th European Conference on e-Government*. The Hague, The Netherlands, 2007
- [KRS07b] KUNIS, R. ; RÜNGER, G. ; SCHWIND, M.: Prozessmappen für das Dokumentenmanagement von Verwaltungsvorgängen. In: GEIST (Hrsg.) ; HEINDL (Hrsg.) ; SCHWEIGHOFER (Hrsg.): *Tagungsband des 10. Internationalen Rechtsinformatik Symposions IRIS 2007*, BOORBERG, 2007
- [Kun07] KUNIS, R.: *Eine Schnittstelle für die Abarbeitung räumlich verteilter Workflows. Grundlagen, Konzeption und Realisierung*. Vdm Verlag Dr. Müller, 2007
- [Lei07] LEITSTELLE INFOHIGHWAY: *InfoHighway Sachsen*. <http://www.infohighway.sachsen.de/index.html>. Version: 2007
- [LSW13] LIPP, M. ; SCHLÜTER, H. ; WEIDAUER, C.: *The Danet Workflow Component - User Manual*. <http://wfmopen.sourceforge.net/userapi/de/danet/an/workflow/api/doc-files/user-manual.html>. Version: 2005 (WfMOpen 1.3)
- [MN05] MENDLING, J. ; NÜTTGENS, M.: EPC Markup Language (EPML) / Vienna University of Economics and Business Administration. Version: März 2005. <http://wi.wu-wien.ac.at/home/mending/publications/TR05-EPML.pdf>. – Technical Report JM-2005-03-10. – Online-Ressource
- [Mom01] MOMJIAN, B.: *PostgreSQL, Einführung und Konzepte*. München : Addison-Wesley, 2001 (Programmer's Choice)
- [Mon04] MONSON-HAEFEL, R.: *Enterprise JavaBeans*. O'Reilly Media, 2004 (4. Auflage)

- [n-k07] N-KOMM GMBH: *e-komm Office*. <http://www.n-komm.de/C1256FE8005A26FB/CurrentBaseLink/W26AKEDR798MONTDE>. Version: 2007
- [Nie06] NIEDERMEIER, S.: *Cocoon 2 und Tomcat. XML-Publishing mit dem Open-Source-Framework*. Bonn : Galileo Press, 2006 (Galileo Computing)
- [Nor02] NORIN, R.: *Workflow Process Definition Interface – XML Process Definition Language*. Version: Oktober 2002. [http://www.wfmc.org/standards/docs/TC-1025\\_10\\_xpd1\\_102502.pdf](http://www.wfmc.org/standards/docs/TC-1025_10_xpd1_102502.pdf). – The Workflow Management Coalition Specification, Abruf: 27. 11. 2007
- [Obj04] OBJECT MANAGEMENT GROUP (OMG): *Common Object Request Broker Architecture: Core Specification – Version 3.0.3*. <http://www.omg.org/docs/formal/04-03-01.pdf>. Version: März 2004
- [OSC01] OSCI-LEITSTELLE: *OSCI: Die informelle Beschreibung*. <http://www.osci.de/materialien/summary.pdf>. Version: November 2001
- [OSC02] OSCI-LEITSTELLE: *OSCI-Transport*. [http://www1.osci.de/sixcms/media.php/13/osci\\_spezifikation\\_1\\_2\\_deutsch.pdf](http://www1.osci.de/sixcms/media.php/13/osci_spezifikation_1_2_deutsch.pdf). Version: April 2002. – Status: Final
- [SAP07] SAPERION AG: *SAPERION Enterprise Solution für Enterprise Content Management (ECM)*. [http://www.saperion.de/de/ecm/produkte/enterprise\\_content\\_management/index.html](http://www.saperion.de/de/ecm/produkte/enterprise_content_management/index.html). Version: 2007
- [Sta04] STARK, T.: *J2EE*. München : Addison-Wesley, 2004 (Master Class)
- [Sun] SUN MICROSYSTEMS: *Java Database Connectivity (JDBC)*. <http://java.sun.com/products/jdbc>
- [TL02] TIDWELL, D. ; LICHTENBERG, K.: *XSLT – XML-Dokumente transformieren*. O Reilly Verlag GmbH & Co., 2002
- [W3C02] W3C: *Zugänglichkeitsrichtlinien für Web-Inhalte 1.0*. <http://www.w3c.de/Trans/WAI/webinhalt.html>. Version: Januar 2002
- [W3C07a] W3C: *Extensible Markup Language (XML)*. <http://www.w3.org/XML/>. Version: Mai 2007
- [W3C07b] W3C: *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*. <http://www.w3.org/TR/soap12-part1/>. Version: April 2007
- [WFM] *Workflow Management Coalition*. <http://www.wfmc.org>
- [WHK97] WAHL, M. ; HOWES, T. ; KILLE, S.: *RFC 2251: Lightweight Directory Access Protocol (v3)*. <ftp://ftp.rfc-editor.org/in-notes/rfc2251.txt>. Version: 1997



## Chemnitzer Informatik-Berichte

In der Reihe der Chemnitzer Informatik-Berichte sind folgende Berichte erschienen:

- CSR-04-01** Karsten Hilbert, Guido Brunnett, A Hybrid LOD Based Rendering Approach for Dynamic Scenes, Januar 2004, Chemnitz
- CSR-04-02** Petr Kroha, Ricardo Baeza-Yates, Classification of Stock Exchange News, November 2004, Chemnitz
- CSR-04-03** Torsten Hoefler, Torsten Mehlan, Frank Mietke, Wolfgang Rehm, A Survey of Barrier Algorithms for Coarse Grained Supercomputers, Dezember 2004, Chemnitz
- CSR-04-04** Torsten Hoefler, Wolfgang Rehm, A Meta Analysis of Gigabit Ethernet over Copper Solutions for Cluster-Networking, Dezember 2004, Chemnitz
- CSR-04-05** Christian Siebert, Wolfgang Rehm, One-sided Mutual Exclusion A new Approach to Mutual Exclusion Primitives, Dezember 2004, Chemnitz
- CSR-05-01** Daniel Beer, Steffen Höhne, Gudula Rünger, Michael Voigt, Software- und Kriterienkatalog zu RAfEG - Referenzarchitektur für E-Government, Januar 2005, Chemnitz
- CSR-05-02** David Brunner, Guido Brunnett, An Extended Concept of Voxel Neighborhoods for Correct Thinning in Mesh Segmentation, März 2005, Chemnitz
- CSR-05-03** Wolfgang Rehm (Ed.), Kommunikation in Clusterrechnern und Clusterverbundsystemen, Tagungsband zum 1. Workshop, Dezember 2005, Chemnitz
- CSR-05-04** Andreas Goerdt, Higher type recursive program schemes and the nested pushdown automaton, Dezember 2005, Chemnitz
- CSR-05-05** Amin Coja-Oghlan, Andreas Goerdt, André Lanka, Spectral Partitioning of Random Graphs with Given Expected Degrees, Dezember 2005, Chemnitz
- CSR-06-01** Wassil Dimitrow, Mathias Sporer, Wolfram Hardt, UML basierte Zeitmodellierung für eingebettete Echtzeitsysteme, Februar 2006, Chemnitz
- CSR-06-02** Mario Lorenz, Guido Brunnett, Optimized Visualization for Tiled Displays, März 2006, Chemnitz
- CSR-06-03** D. Beer, S. Höhne, R. Kunis, G. Rünger, M. Voigt, RAfEG - Eine Open Source basierte Architektur für die Abarbeitung von Verwaltungsprozessen im E-Government, April 2006, Chemnitz
- CSR-06-04** Michael Kämpf, Probleme der Tourenbildung, Mai 2006, Chemnitz
- CSR-06-06** Torsten Hoefler, Mirko Reinhardt, Torsten Mehlan, Frank Mietke, Wolfgang Rehm, Low Overhead Ethernet Communication for Open MPI on Linux Clusters, Juli 2006, Chemnitz

## Chemnitzer Informatik-Berichte

- CSR-06-07** Karsten Hilbert, Guido Brunnett, A Texture-Based Appearance Preserving Level of Detail Algorithm for Real-time Rendering of High Quality Images, August 2006, Chemnitz
- CSR-06-08** David Brunner, Guido Brunnett, Robin Strand, A High-Performance Parallel Thinning Approach Using a Non-Cubic Grid Structure, September 2006, Chemnitz
- CSR-06-09** El-Ashry, Peter Köchel, Sebastian Schüler, On Models and Solutions for the Allocation of Transportation Resources in Hub-and-Spoke Systems, September 2006, Chemnitz
- CSR-06-10** Raphael Kunis, Gudula Rünger, Michael Schwind, Dokumentenmanagement für Verwaltungsvorgänge im E-Government, Oktober 2006, Chemnitz
- CSR-06-11** Daniel Beer, Jörg Dümmler, Gudula Rünger, Transformation ereignisgesteuerter Prozeßketten in Workflowbeschreibungen im XPDL-Format, Oktober 2006, Chemnitz
- CSR-07-01** David Brunner, Guido Brunnett, High Quality Force Field Approximation in Linear Time and its Application to Skeletonization, April 2007, Chemnitz
- CSR-07-02** Torsten Hoefler, Torsten Mehlan, Wolfgang Rehm (Eds.), Kommunikation in Clusterrechnern und Clusterverbundsystemen, Tagungsband zum 2. Workshop, Februar 2007, Chemnitz
- CSR-07-03** Matthias Vodel, Mirko Caspar, Wolfram Hardt, Energy-Balanced Cooperative Routing Approach for Radio Standard Spanning Mobile Ad Hoc Networks, Oktober 2007, Chemnitz
- CSR-07-04** Matthias Vodel, Mirko Caspar, Wolfram Hardt, A Concept for Radio Standard Spanning Communication in Mobile Ad Hoc Networks, Oktober 2007, Chemnitz
- CSR-07-05** Raphael Kunis, Gudula Rünger, RAfEG: Referenz-Systemarchitektur und prototypische Umsetzung - Ausschnitt aus dem Abschlussbericht zum Projekt "Referenzarchitektur für E-Government" (RAfEG) -, Dezember 2007, Chemnitz