

Folie 0

Der folgende Vortrag soll umfassend informieren und ist etwas länger, aber dadurch werden viele zu erwartende Fragen beiläufig beantwortet.

Folie 1

Bevor ich mit meinem Vortrag beginne, kurz etwas zu meiner Person.

Mein Name ist Frank Obermeit und ich bin von Beruf Dipl.-Informatiker. In Rechenzentren arbeitete ich 35 Jahre und habe die Datenverarbeitung von der Pike auf gelernt. Seit 2011 bin ich im Landesarchiv Sachsen-Anhalt als Informatiker tätig und beschäftige mich u.a. mit der Automation von Abläufen bei der Verwaltung von Digitalisaten und Digitalen Objekten.

Folie 2

In meinem Impulsvortrag möchte ich mich mit der Frage auseinandersetzen „ob es möglich ist, eine OAIS-konforme Software-Architektur, basierend auf REST und BPMN, zu entwickeln?“.

Folie 3

Die Kernaussage meines Vortrages kann in einer Folie zusammengefasst werden.

Die Archival Storages werden mit REST gekapselt, die Prozessmodellierung und Prozessverarbeitung erfolgt mit BPMN, alles wird über eine webbasierte Benutzeroberfläche dem Nutzer zur Verfügung gestellt und die vorhandene Rechteverwaltung wird integriert.

Wie ich zu dieser Konzeption kam, möchte ich Ihnen mit dem folgenden Vortrag erläutern.

Folie 4

Die Hauptaufgabe eines Rechenzentrums besteht in der Automation von Prozessen und Abläufen. Seit 1976 beschäftige ich mich dadurch unweigerlich mit der Prozessautomation, die zu meinen Lieblingsthemen zählt.

Noch vor zehn Jahren blieb Fähigkeit der Prozessautomation dem Hostbereich vorbehalten, da die angebotenen Produkte wegen der horrenden Kosten im kleineren Rahmen nicht einsetzbar waren.

Oracle bot beispielsweise ein datenbankgestütztes Scheduling<sup>1</sup> an, aber es hätte „native“ nie in einem größeren Umfang eingesetzt werden können.

---

<sup>1</sup> <https://docs.oracle.com/en/database/oracle/oracle-database/12.2/admin/oracle-scheduler-concepts.html#GUID-6B8C0335-D36B-44D3-95BC-82FA2141D967>

Die Zeiten haben sich geändert.

Die Prozessautomation lässt sich heute im kleinen Rahmen mit Open Source- Produkten realisieren und hier liegt einer der Schlüssel für die software-technische Umsetzung des OAIS-Referenzmodells.

Folie 5

Meinen weiteren Ausführungen möchte ich unbedingt vorausschicken, dass das DIMAG damit nicht infrage gestellt wird. DIMAG ist ein Projekt und in dessen Rahmen entstehende Konzepte und Produkte sind die Ergebnisse der föderalen Zusammenarbeit.

Mein Vortrag soll ein Impuls sein, mit dem ich dem DIMAG einen Schub geben und es einen wesentlichen Schritt voranbringen möchte.

Diese universelle Methode kann auch in anderen Umgebungen eingesetzt werden, aber da sich viele Archive für DIMAG entschieden haben, würde die dortige Umsetzung die größte Wirkung erzielen.

Mit dieser Aussage möchte ich der Vermutung entgegentreten, dass sich das Landesarchiv Sachsen-Anhalt vom DIMAG abwenden möchte.

Folie 6

Beginnen werde ich den Vortrag mit dem Ihnen vertrauten OAIS-Referenzmodell und Sie damit in Ihrem Fachgebiet abholen.

Anschließend werde ich Sie an die konzeptionelle und software-technische Lösung heranzuführen, die ich mir vor 18 Monaten überlegt, im Laufe des letzten Jahres konzipiert und mittels Prototyping auf ihre Machbarkeit untersucht habe, denn Sie wollen Lösungen und keine graue Theorie.

Bereits an dieser Stelle möchte ich Ihnen sagen, dass diese Lösung auf Open Source-, Community- Produkten und „no-cost-features“ basiert und keine zusätzlichen Kosten verursacht.

Sie können natürlich auch die kostenpflichtigen Enterprise-Versionen einsetzen.

Folie 7

Zeitlicher Projektverlauf:

- |                 |   |
|-----------------|---|
| 2017/II         | Speicherkonzept des Landesarchiv Sachsen-Anhalt   |
| 2017/III        | BPMN als möglicher Lösungsansatz für Workflowsteuerung  |
| 2017/III u. IV. | Suche nach technischen Komponenten für eine ganzheitliche Verwaltung von Digitalisaten und Digitalen Objekten |

Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

2018/01	Vorstellung einer ersten Idee einer OAIS-konformen-Software-Architektur innerhalb des DAN und DIMAG
2018/02 - 05	Einarbeitung in relevante Technologien und Produkte
2018/06	Erster Prototyp zum Konzept fertig – aber es bestand ein erhöhter und inakzeptabler Programmieraufwand bei der Benutzeroberfläche
2018/08	Benutzeroberflächen-Komponente in neuer Version verfügbar und die vorher inakzeptable Programmierung war überflüssig Machbarkeit wurde wiederholt nachgewiesen
2018/09	Vorstellung des Konzeptes und weitere prototypische Umsetzung im Landesarchiv Sachsen-Anhalt
2018/11	Vorstellung des Konzeptes inkl. Prototyp beim DIMAG-Entwicklertreffen

Von der Idee bis zum fertigen Konzept inkl. Prototyp vergingen nur zwölf Monate.

Folie 8

Nochmals zum Thema des Vortrages.

Ist es möglich, eine OAIS-konforme Software-Architektur, basierend auf REST und BPMN, zu entwickeln?

Folie 9

Es ist demnach die Frage zu beantworten: Welche software-technischen Komponenten werden für die Umsetzung der OAIS-Funktionseinheiten benötigt?

Folie 10 und 11

Meine Motivation, sich mit diesem Thema zu beschäftigen, resultierte aus der Anforderung, dass im Landesarchiv Sachsen-Anhalt ein dreistufiges Speicherkonzept entwickelt und umgesetzt werden sollte.

Folie 12

Der damalige Anspruch war, eine ganzheitliche Lösung für Digitalisate und genuin digitale Objekte zu konzipieren, die sich an etablierten Standards und Technologien orientiert.

Es sollten alle Prozesse außerhalb des „Digitalen Massenspeichers“, dem DIMAG-Kernmodul, einbezogen werden.

Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

Archivaren sollte die Möglichkeit einer freien Gestaltung der Benutzeroberfläche geboten werden und Workflows sollten graphisch modellierbar sein.

In BPMN sah ich eine Möglichkeit, eine hohe Flexibilität bei geringem Programmieraufwand zu erreichen.

Die Rechteverwaltung hatte bei der Auswahl der Produkte immer die höchste Priorität und ein Produkt hätte nie ohne diese eingesetzt werden können.

Folie 13 - ausgeblendet

Der damalige Zeitplan sah eine Einarbeitung in alle notwendigen Basistechnologien und Standards vor.

Wobei Docker, als Container-Virtualisierung, die Basis für ein schnelles Prototyping im Sinne von DevOps (Development and Operations) bilden sollte.

Das Konzept sollte bis zum Juni erarbeitet und im Januar 2019 abgeschlossen werden, um es dann auf der AUdS- Tagung im März 2019 vorstellen zu können.

Folie 14

Heute möchte ich Ihnen die Ergebnisse präsentieren.

Folie 15

Zunächst werde ich Sie, wie bereits angekündigt, beim funktionalen **OAIS-Referenzmodell** abholen und rhetorisch fragen, ob sich Workflows hinter den einzelnen Funktionseinheiten verbergen.

Folie 16

Beginnen wir mit der Funktionseinheit **Ingest bzw. Übernahme**.

Folie 17

Kurz eingeblendet die Rollenbeschreibung dieser Funktionseinheit – nachzulesen in der deutschen Übersetzung des OAIS-Referenzmodells.

Folie 18

Kernaufgaben dieser Funktionseinheit sind die Übernahmen der SIPs und die Bereitstellung angemessener Speichermöglichkeiten.

Die QS, das Erzeugen der AIPs, das Extrahieren von Erschließungs- informationen und die Koordination der Aktualisierungen des Archivspeichers und der Datenverwaltung sind Prozesse, die mit BPMN modelliert und teilweise automatisiert werden können.

Folie 19 – nur Übergang

Folie 20

Betrachten wir die Funktionseinheit **Preservation Planning bzw. Erhaltungsplanung**

Folie 21

Hier wieder die Rollenbeschreibung dieser Funktionseinheit kurz eingeblendet.

Folie 22

Die Kernaufgaben bilden modellierbare und automatisierbare Prozesse, aber auch konzeptionelle Aufgaben z.B. Empfehlungen für Standards und Richtlinien entwickeln.

Folie 23 – nur Übergang

Folie 24

Betrachten wir das **Data Management bzw. die Datenverwaltung**

Folie 25

Hier wieder die Rollenbeschreibung dieser Funktionseinheit kurz eingeblendet.

Folie 26

Wie beim Preservation Planung gibt es hier modellierbare und automatisierbare Prozesse, aber auch konzeptionelle Aufgaben.

Folie 27 – nur Übergang

Folie 28

Betrachten wir die Funktionseinheit **Access bzw. Zugriff**

Folie 29

Hier wieder die Rollenbeschreibung dieser Funktionseinheit kurz eingeblendet.

Folie 30

Hier dominieren wieder modellierbare und automatisierbare Prozesse. Insbesondere, weil der Zugriff klar geregelt und dokumentiert werden muss.

Der Zugriffschutz wird natürlich über eine Rechteverwaltung gewährleistet.

Folie 31 – nur Übergang

Folie 32

Betrachten wir die **Administration**.

Folie 33

Hier wieder die Rollenbeschreibung dieser Funktionseinheit kurz eingeblendet.

Folie 34

Das Konzeptionelle steht im Vordergrund und dafür soll auf Reports zurückgegriffen werden, deren Erstellung automatisiert werden kann.

Folie 35 - ausgeblendet

Folie 36

Bis hier kann festgehalten werden, dass jede Funktionseinheit des Modells Funktionen enthält, die sich teilweise modellieren und automatisieren lassen.

Die Automation kann mit **BPMN** modelliert und wie wir später sehen werden, auch für die tatsächliche Prozesssteuerung eingesetzt werden.

Folie 37

Zunächst einige kurze Begriffserklärungen, bevor wir uns der Funktionseinheit Archival Store bzw. Archivspeicher widmen.

**BPMN** - Akronyme - kurz erklärt

Business Process Model and Notation (BPMN<sup>2</sup>)

Wird für die graphische Modellierung vorhersehbarer Prozesse benutzt.

Decision Model and Notation (DMN<sup>3</sup>)

Wird für die graphische und tabellarische Modellierung von Entscheidungstabellen benutzt.

Case Management Model and Notation (CMMN<sup>4</sup>)

CMMN spielt momentan keine Rolle, da sie für die Modellierung nicht so klar spezifizierbarer Prozessabläufe genutzt wird.

BPMN und DMN werden Sie später in der Modellierung und Benutzung sehen.

Folie 38

**REST**

---

<sup>2</sup> [https://de.wikipedia.org/wiki/Business\\_Process\\_Model\\_and\\_Notation](https://de.wikipedia.org/wiki/Business_Process_Model_and_Notation)

<sup>3</sup> [https://de.wikipedia.org/wiki/Decision\\_Model\\_and\\_Notation](https://de.wikipedia.org/wiki/Decision_Model_and_Notation)

<sup>4</sup> <https://en.wikipedia.org/wiki/CMMN>

Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

REST (seltener ReST) steht für Representational State Transfer und bezeichnet ein Programmierparadigma für verteilte Systeme, insbesondere für Webservices (siehe [wiki](#)<sup>5</sup>).

REST wird, neben SOAP (Simple Object Access Protocol), vorrangig für die Maschine-Maschine-Kommunikation eingesetzt.

Der Begriff RESTful oder RESTful HTTP taucht häufig auf und beschreibt lediglich, dass REST unter Ausnutzung der Möglichkeiten des Application Protocol HTTP umgesetzt wird.

Im Laufe des Vortrages wird die Bedeutung von REST klarer.

Folie 39

Betrachten wir also den **Archival Storage bzw. Archivspeicher**.

Folie 40

Hier wieder die Rollenbeschreibung dieser Funktionseinheit kurz eingeblendet

Folie 41

Die Kernaufgaben des Archivspeichers bilden modellierbare und automatisierbare Prozesse.

Folie 42

Der Archivspeicher ist die zentrale Funktionseinheit des Modells, denn die Daten sollen dauerhaft erhalten bleiben und müssen jederzeit interpretierbar sein.

Somit sollten die Metadaten und die eigentlichen Daten immer zusammen aufbewahrt werden.

Hier bietet sich die Kapselung durch ein **REST-basiertes API** an.

Erstmals spielen **REST und BPMN** eine sichtbare Rolle.

Folie 43

Betrachten wir zunächst **REST für den Archivspeicher** und gehen danach auf **BPMN** ein.

Folie 44

Die Funktionseinheiten **Ingest, Access und Administration** nutzen laut OAIS-Referenzmodell den Archival Storage **direkt**.

**Preservation Planning und Data Management indirekt**.

---

<sup>5</sup> [https://de.wikipedia.org/wiki/Representational\\_State\\_Transfer](https://de.wikipedia.org/wiki/Representational_State_Transfer)

Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

Beim Ingest besteht die Besonderheit, dass dieser ebenfalls Storages für die SIPs benötigt.

Folie 45 – 48 - Animation

Der Archivspeicher dient der dauerhaften Aufbewahrung von AIPs.

Beim Ingest wird ebenfalls ein Storage(s) für SIPs benötigt.

Was liegt näher, diesen ebenfalls als „Archivspeicher“ zu bezeichnen und ihn wie diesen zu konzipieren.

Folie 49

Storages müssen adressierbar sein und dafür bieten sich zunächst die **UUIDs** als eindeutige Namen an, deren Lokationen durch Services aufgelöst werden sollten.

Folie 50

**URI, URN und URL** fallen einem dabei sofort ein.

Folie 51

Ein **RESTful-Webservice** könnte mit den Möglichkeiten von HTTP die eigentlichen Daten und Metadaten **kapseln**.

Mit RESTful-Webservices kann der Zugriff auf die in Storages gespeicherten Daten und Metadaten gekapselt werden.

Folie 52 ausgeblendet

Folie 53

RESTful-Webservices können mit etablierten Technologien und Produkten umgesetzt werden und damit kann für die Aufgaben Authentication, Authorization, Load Balancing und Access fertige Lösungen zurückgegriffen werden.

Die eigentliche Kapselung der Daten muss bei Web-Servern nur konfiguriert werden.

Es gibt viele Command Line Interface (CLI) - Tools.

Als letzte zu Aufgabe verbleibt die Implementierung der Funktionen – sprich Web-Service-Operationen.

Hier liegt die besondere Bedeutung von REST, denn für die Authentifikation, den Zugriffsschutz und das Load-Balancing können die „normalen“ Web-Technologien und Produkte eingesetzt werden.

SOAP und REST sind Batch-fähig, können in Web-Anwendungen und standalone Programmen genutzt werden.

## Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

Auf den nachfolgenden Folien wird die Kapselung der Storages für Digitalisate und Digitale Objekte mit RESTful- WebServices gezeigt.

Bisher haben Sie nur graue Theorie gehört und es wird Zeit die tatsächliche Benutzung und einfache Handhabung zu demonstrieren.

### Folie 54

Bei der Programmierung oder deren Beauftragung werden Sie vom Open-Source-Framework swagger unterstützen. Es unterstützt Sie beim Entwurf, der Erstellung, Dokumentation und Nutzung von RESTful- Webservices. Hier sehen Sie die interaktiv benutzbare und zur Laufzeit von swagger generierte Dokumentation, die Sie im Browser verwenden.

Hier sehen Sie eine mit swagger zur Laufzeit generierte und interaktiv benutzbare Dokumentation.

Die Kapselung des Datenzugriffs und der Datenbearbeitung über RESTful- Webservices bedeutet zwar Programmieraufwand, allerdings werden der Entwurf, die Erstellung, die Dokumentation und die Nutzung von WebServices durch den Einsatz des Open-Source-Framework Swagger wesentlich vereinfacht.

In der Konsequenz bedeutet dies, dass lediglich die Implementierung von RESTful- Webservices beauftragt werden muss und nicht die Workflows, die laufenden Änderungen unterliegende.

Hier ein Read Only API am Beispiel eines Service, der die DROID-Format- Identifikation kapselt.

### Folie 55

Hier ein Beispiel für einen RESTful-Webservice, der neben GET auch die Methoden PUT, POST und DELETE nutzt.

**POST:** Fügt eine neue (Sub-)Ressource unterhalb der angegebenen Ressource ein. Da die neue Ressource noch keinen URI besitzt, adressiert der URI die übergeordnete Ressource. Als Ergebnis wird der neue Ressourcenlink dem Client zurückgegeben.

**POST** kann im weiteren Sinne auch dazu verwendet werden, Operationen abzubilden, die von keiner anderen Methode abgedeckt werden.

**GET:** Wird für Reads benutzt.

**PUT:** Die angegebene Ressource wird angelegt. Wenn die Ressource bereits existiert, wird diese geändert.

## Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

Im Zusammenhang mit RESTful wird der Begriff **Idempotent** benutzt und dies bedeutet, dass laut HTTP-Spezifikation für die Methoden GET, HEAD, PUT und DELETE gilt:

Das mehrfache Absenden der gleichen Anforderung wirkt sich **nicht** anders als ein einzelner Aufruf aus (siehe wiki<sup>6</sup>).

### Folie 56

Ein weiteres Beispiel aus dem Bereich der Digitalisate, bei dem UUID zur Identifikation von Storages und Daten eingesetzt werden.

Unterhalb eines Storage könnten weitere Dateneinheiten adressiert werden, deren Granularität vom Anwendungsfall abhängt.

Wenn Sie konsequent mit **URIs** arbeiten, dann können Sie, wie Sie später noch sehen werden, Workflows konzipieren, die eine Datenverarbeitung und Datenbearbeitung veranlassen, ohne die Daten zu bewegen.

### Folie 57

Das nachfolgende Video zeigt die interaktive Benutzung, der vom Swagger-Framework zur Laufzeit erzeugte Dokumentation eines RESTful-WebServices. Interessant ist daran, dass sich jeder Archivar oder Entwickler einen Funktionsüberblick verschaffen kann und die Beauftragung einer Schnittstellen-Implementierung elegant beschrieben und eindeutig spezifiziert werden kann.

Ablauf: Download der für DROID notwendigen Signatur-Datei, Formatidentifikation für eine mit einer URI adressierte Ressource und anschließend die Formatidentifikation für eine mit einer URL adressierte Ressource. Für schnelle Tests wird eine URI bzw. URL generiert, wenn kein Wert für den Parameter URI oder URL eingetragen wird. Daher wird jede Operation mit einer dann gültigen URI bzw. URL wiederholt.

### Folie 58

Hervorzuheben ist, dass ein RESTful-Webservice während der Entwicklung beiläufig dokumentiert wird und Sie diesen Webservice mit swagger ohne großen Aufwand testen können. Hier sei nochmals betont, dass RESTful-WebServices ein elegantes Mittel zur Kapselung beliebiger Funktionalitäten sind.

Die Informationen für die von swagger zu generierende Dokumentation werden in sogenannten Annotation hinterlegt.

Im Laufe des Vortrages werde ich Ihnen die Verknüpfung von Webservices mittels BPMN zeigen. Dafür habe ich exemplarisch drei RESTful-Services

---

<sup>6</sup> [https://de.wikipedia.org/wiki/Representational\\_State\\_Transfer](https://de.wikipedia.org/wiki/Representational_State_Transfer)

Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

implementiert, aber nur zwei werden in den noch zu sehenden Videos benutzt.

Hier sehen die Implementierung mit der Integrierten Entwicklungsumgebung Eclipse. Enthalten sind die swagger-Annotationen für die Generierung der Dokumentation und die eigentliche Kapselung, die effektiv vier Zeilen Java-Code umfasst.

Folie 59

Halten wir fest: Die erste software-technische Komponente sind die Storages, die mit RESTful-Webservices gekapselt werden können.

0..n meint dabei, dass Storages existieren können.

Folie 60

Wenden wir uns jetzt BPMN zu.

Folie 61

Was ist die Herausforderung?

Folie 62

Alle Funktionseinheiten beinhalten modellierbare und automatisierbare Prozesse und das Dilemma besteht darin, dass Sie eine Möglichkeit zur freien Modellierung suchen, denn Sie wollen nicht für jeden Prozess(ablauf) einen neuen Projektauftrag auslösen und einen externen Dienstleister bemühen.

Folie 63

Die Möglichkeiten von BPMN möchte ich Ihnen mit dem nachfolgenden Video verdeutlichen.

Folie 64

Dabei wird ein zweistufiges und ggf. iteratives Vorgehen empfohlen. Das zunächst fachliche Modell wird für die technische Umsetzung ergänzt und modifiziert. Sie arbeiten aber immer mit einem Modell (Dokument) von der fachlichen Beschreibung bis zur technischen Abarbeitung. Das ist die Stärke von BPMN, denn es gibt keinen Medienbruch zwischen dem Fach- und dem IT-Bereich.

Folie 65

Hier sehen Sie das **gesamte BPMN-Modell** auf einen Blick.

Es soll der Prozess von der Anlieferung des SIP durch die abgebende Stelle, per FTP, bis zur Übernahme in den Digitalen Langzeitspeicher DIMAG modelliert und realisiert werden.

Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

Der Workflow beginnt mit der Übergabe des SIP per FTP, anschließend wird der Mitarbeiter oder eine Gruppe per E-Mail darüber informiert. Automatisch wird die Format-Validierung mit JHove durchgeführt, die Gruppe oder ein Mitarbeiter werden wieder informiert. Ein Mitarbeiter kann dann das SIP für die AIP-Erstellung vorbereiten. Danach wird das AIP erzeugt, per SFTP ins DIMAG übernommen und ein Mitarbeiter importiert das AIP im DIMAG.

Das nachfolgende Video zeigt die Prozessmodellierung nach BPMN mit dem Open-Source-Modeller von der Berliner Firma Camunda im Zeitraffer – zehnfache Geschwindigkeit.

Folie 66 und 67 – ein- und zweiminütiges Video

Da in dem Prozess eine Entscheidungstabelle benötigt wird zeige ich Ihnen erst die Modellierung einer Entscheidungstabelle und dann die Modellierung des BPMN-Prozesses.

Folie 68

BPMN ist eine sehr mächtige, aber gut verständliche Modellierungssprache. Sie haben in dem Video gesehen, dass Aktivitäten als Tasks modelliert und typisiert wurden. Zusammenfassen würde ich es so formulieren:

1. Sie kombinieren Tasks miteinander.
2. Die Abarbeitungsreihenfolge kann über Gateways gesteuert werden.
3. Der Prozess kann auf Ereignisse reagieren, die ebenfalls modelliert werden können.

Bei den Tasks wird im Wesentlichen zwischen User-, Manual, Service-, Skript und Business Rules Tasks unterschieden.

**Send- und Receive-Task** sind selbsterklärend.

**User-Tasks** bilden Aufgaben ab, die von einem Menschen ausgeführt werden. Die dabei anfallenden Daten können dabei in den Prozess einfließen. Daneben gibt es noch die **Manual-Tasks**, die für die Modellierung externer Aufgaben genutzt werden.

In diesem Fall soll beim Start dem Prozess die URI des SIP, also der Bagit-Tar-Datei, übergeben werden.

Eine **Script-Task** dient der automatischen Verarbeitung.

Natürlich können auch externe Programme eingebunden werden, aber oftmals genügt die Abarbeitung von Skripten bzw. Skript-Anweisungen.

Daneben gibt es noch die **Service-Task**, mit der Geschäftslogik aufgerufen oder ausgeführt werden kann. Diese werden beispielsweise für die Benutzung der REST-Services genutzt.

Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

Hinter den **Business-Rule-Tasks** verbergen sich mit DMN modellierte Entscheidungstabellen.

Folie 69

Nachdem dieses Modell entworfen wurde, können technische Parameter den **Tasks** und **Gateways** hinzugefügt werden.

Anschließend kann es einer BPMN-Engine zur Abarbeitung übergeben werden.

Auf den nachfolgenden Folien zeige ich Ihnen die wesentlichen programmtechnischen-Ergänzungen, die keine besonderen Programmierkenntnisse erfordern und auch von Archivaren vorgenommen werden können.

Wie bereits erwähnt, ein großer Nutzen besteht darin, dass die fachlichen und technischen Informationen in einem BPMN-Modell, spricht in einer Datei, dokumentiert werden. Der Dokumentationswert ist ungemein hoch.

Folie 70 - 71

Es wird geprüft, ob die Variable „bagitTarUri“ vorhanden ist.

Folie 72 - 74

Die weitere Verarbeitung nach dem XOR-Gateway ist vom vorherigen Auswertungsergebnis abhängig.

Folie 75 - 77

Ein Mitarbeiter erhält eine E-Mail, wofür der mail-Connector zu konfigurieren ist. Dieser Connector bietet sich für den E-Mail-Versand an, da er mit geringem Aufwand dynamisch konfiguriert werden kann.

Folie 78 - 80

In diesem Task wird erstmalig ein RESTful-Webservice über den http-Connector genutzt. In diesem Fall wird das Tar-File entpackt und eine URI des entpackten Bagit als Variable in den Prozess aufgenommen.

Folie 81 - 83

Ein RESTful-Webservice wird für die JHove-Format-Validierung konfiguriert. Sechs Zeilen Javascript mussten programmiert werden.

Folie 84 - 87

Die Ein- und Ausgabeparameter werden für Entscheidungstabelle konfiguriert.

Folie 88 - 90

## Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

Das Ergebnis der Entscheidungstabelle wird als Variable in den Prozess aufgenommen.

Folie 91 - 93

Der Programmieraufwand für die Zusammenführung aller JHove-Ergebnisse umfasst 26 Zeile Javascript.

Die Konfiguration setzt sich so fort. Das Vorgehen wiederholt sich:

- Nutze einen Webservice,
- werte Variablen aus und
- erzeuge neue Variablen.

Folie 94 - 97

An einer Stelle musste ich auf Groovy zurückgreifen, weil ich mit Javascript nicht die notwendige Variable für die Iteration erzeugen konnte. Groovy ist eine Skript- und Programmiersprache, die auf JVMs ausgeführt werden kann und damit für viele Plattformen verfügbar ist.

Folie 98

Die technische Konfiguration ist für viele Archivare wahrscheinlich keine Herausforderung. Die Abarbeitung kann durch die Auswertung von Daten (Prozessvariablen und anderen Daten) dynamisch gestaltet werden. Es werden nur Metadaten im Prozess bewegt und die eigentlichen Daten können unverändert an ihrem Ort verbleiben. Die eingesetzte BPMN-Engine bietet einen REST- und einen SOAP-Connector an.

Folie 99

Nachdem der Workflow vollständig modelliert und alle technischen Parameter konfiguriert wurden, kann das BPMN-Modell der Engine zur Abarbeitung übergeben werden. Das Deployment erfolgt direkt aus dem BPMN-Modeller heraus.

Folie 100 - ausgeblendet

Folie 101

Die zweite software-technische Komponente heißt BPMN und 1..n+1 steht für eine allgemeine und alle Storages einbeziehende Komponente, die natürlich über REST gekapselt wird.

Folie 102

Die Benutzeroberfläche ist eine austauschbare Komponente und soll die Benutzung der beiden Komponenten Storages und BPMN erleichtern.

Entscheidend ist die ausschließliche Nutzung der RESTful-Webservices von Storages und BPMN.

Folie 103

Jede dieser Komponenten wurde prototypisch implementiert.

Folie 104

Die vollständige Abarbeitung des eben modellierten Prozesses vom FTP-Upload bis zum Upload ins DIMAG sehen Sie im nachfolgenden Video. Die nutzerfreundliche Integration von BPMN und RESTful-Webservices erfolgte mit Application Express von Oracle, womit die Anwendung vollständig Web-basiert ist. SSO und die LDAP-Authentifikation wurden für Camunda aus Zeitmangel noch nicht konfiguriert.

Folie 105 – zweiminütiges Video

Folie 106

Leider konnte ich nicht alle Möglichkeiten mit Videos veranschaulichen, aber ich möchte Sie auf weitere sehr wesentliche Vorteile von BPMN und DMN aufmerksam machen, deren Potentiale Sie wahrscheinlich sofort erkennen werden. Hervorheben möchte ich:

1. BPMN und RESTf in standalone Anwendungen
2. BPMN und REST für verteilte Transaktionen
3. Decision Model and Notation – DMN

Folie 107

Auch bei standalone Programmen gilt, die Anwendungssteuerung wird mit BPMN modelliert und darüber werden die Funktionalitäten miteinander kombiniert.

Programme verketteten Funktionalitäten miteinander. Ein großer Vorteil von BPMN besteht darin, dass Sie es auch innerhalb von standalone Programmen anwenden können.

Folie 108

Dadurch müssen nur die technischen Funktionalitäten implementiert werden und der eigentliche Verarbeitungsprozess kann jederzeit von Ihnen angepasst werden.

Folie 109

Mit BPMN können Sie verteilte Transaktionen modellieren und de facto implementieren. Ein Rollback kann bei verteilten Transaktionen nur über eine implementierte Kompensation durchgeführt werden und BPMN ist dafür ideal geeignet.

Folie 110

Auch hier gilt, Funktionalitäten lassen Sie implementieren, aber die Prozesssteuerung, also die Kompensation, gestalten Sie.

Folie 111

DMN (Decision Model an Notation) entlastet Sie von der Implementierung von Entscheidungstabelle (Business Rules), die technisch anspruchsvoll und meist unflexibel ist.

Folie 112 und 113

Entscheidungstabellen können Sie graphisch modellieren und miteinander, auch kaskadiert, kombinieren. Sie sind in BPMN-Modelle integrierbar und bieten damit eine weitere nicht zu unterschätzende Flexibilität bei der Prozessgestaltung.

Folie 114

BPMN, kombiniert mit DMN, beschränkt die Programmierung auf ein Mindestmaß und Prozessabläufe werden damit jederzeit änderbar.

Folie 115

BPMN kombiniert mit DMN

- Fachliche Anforderungen können außergewöhnlich schnell umgesetzt werden.
- Es wird ein Höchstmaß an Flexibilität bei der Prozessmodellierung erreicht.
- Produktabhängigkeiten werden auf das Wesentliche reduziert.
- Funktionale Produkte sind austauschbar.

Folie 116

Es bleibt noch die Frage zu beantworten: Welche software-technischen Komponenten werden für die Umsetzung der OAIS-Funktionseinheiten benötigt?

Folie 117

Es wurde ein fiktiver Anwendungsfall aus dem Bereich der genuin Digitalen Daten modelliert. Dabei wurde deutlich, wie schnell eine Workflow-Beschreibung modelliert und überprüft werden kann und die Zusammenarbeit zwischen dem Fachbereich und den „Technikern“ erleichtert wird.

Folie 118

Mit BPMN und REST lässt sich eine OAIS-konforme Software-Architektur konzipieren und umsetzen.

Diese Methode kann auf alle Funktionsbereiche des OAIS-Modells angewendet werden.

Folie 119

## Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

Warum BPMN?

Folie 120

BPMN ist anwendbar vom Modell bis zur Abarbeitung

BPMN beschreibt den Workflow fachlich und technisch

BPMN ist eine intuitive Modellierungssprache

BPMN ermöglicht ein schnelles Prototyping von der Anforderung bis zur Umsetzung

BPMN wird mit DMN ergänzt

Business Rules - Entscheidungstabellen

CMMN ist noch nicht relevant

Folie 121

Warum REST?

Folie 122

REST ermöglicht produktunabhängige Modellierungen der Storages

REST ist anwendbar für Ingest- und Archival-Storages

REST ist gut dokumentierbar und historisierbar

REST ermöglicht schnelle Produktentwicklungen und verkürzt die Entwicklungszyklen

Folie 123

Wurden die gestellten Ansprüche erfüllt?

Der damalige Anspruch war, eine ganzheitliche Lösung für Digitalisate und genuin digitale Objekte zu konzipieren, die sich an etablierten Standards und Technologien orientiert.

Archivaren sollte die Möglichkeit der freien Gestaltung der Benutzeroberfläche geboten werden und Workflows sollten graphisch modellierbar sein.

BPMN bietet eine hohe Flexibilität, beschränkt den Programmieraufwand auf ein Mindestmaß.

Alle Produkte bieten die zwingend erforderliche Rechteverwaltung.

Folie 124

Wenn Sie nach dieser Methode vorgehen, dann können Sie sich auf die fachliche Modellierung konzentrieren.

Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

Durch den Einsatz von BPMN und REST werden Ihre Prozesse ausreichend dokumentiert.

Sie nutzen Standards und benötigen keine Eigenentwicklungen.

Sie setzen professionelle Produkte ein.

Sie setzen ein einheitliches Framework ein, das Sie aber individuell nutzen.

Folie 125 - ausgeblendet

Folie 126 – Der Vortrag in einer Folie

Folie 127 - ausgeblendet

Folie 128

Meine Hauptmotivation war, dass ich mit dieser Arbeit einen entscheidenden Beitrag im Bereich der Archivierung Digitaler Objekte leisten und einen Fußabdruck im Archivwesen hinterlassen wollte.

Folie 129

Diese nicht ganz ernstgemeinte Folie wollte ich Ihnen nicht vorenthalten.

Folie 130

Kontaktdaten

Frank Obermeit

Landesarchiv Sachsen-Anhalt

Brückstraße 2

39114 Magdeburg

0391 59806 224

[frank.obermeit@la.sachsen-anhalt.de](mailto:frank.obermeit@la.sachsen-anhalt.de)

[frank.obermeit@t-online.de](mailto:frank.obermeit@t-online.de)

Folie 131

Die Authentifikation wird gegen ein LDAP z.B. Active Directory mit SSO durchgeführt.

Das Framework wird als Web-Anwendung über einen Apache HTTP-Server bereitgestellt.

REST wird als Kommunikations-Schnittstelle benutzt.

## Mit BPMN und REST zu einer OAIS-konformen Software-Architektur

Camunda BPM-Engine benötigt eine beliebige Datenbank, wenn die Prozessdaten dauerhaft aufbewahrt werden sollen. Die Benutzung der Engine erfolgt über eine Tomcat-Web- Anwendung.

APEX, eine low-code development platform und ein no-cost-feature von Oracle, wird für die Entwicklung der Web-Oberfläche eingesetzt. ORDS, ebenfalls eine no-cost-Option von Oracle, stellt selbst RESTful-Web-Services bereit, wird aber als Listener für APEX genutzt und als Web-Applikation in den Tomcat-Web-Server eingebunden.

Eine Oracle-Datenbank ist für APEX zwingend erforderlich.

Folie 132

Es werden nur Open-Source-Produkte bzw. Oracle-no-cost-options eingesetzt.

Weitere Produkte für den RESTful-Webservice

- BagIt Library (BIL)<sup>7</sup>
- file format identification tool <sup>8</sup>
- org.openpreservation.jhove<sup>9</sup>

---

<sup>7</sup> <https://github.com/LibraryOfCongress/bagit-java>

<sup>8</sup> <http://www.nationalarchives.gov.uk/information-management/manage-information/preserving-digital-records/droid/>

<sup>9</sup> <http://jhove.openpreservation.org>