



Kopplung von Arcinsys und Kitodo

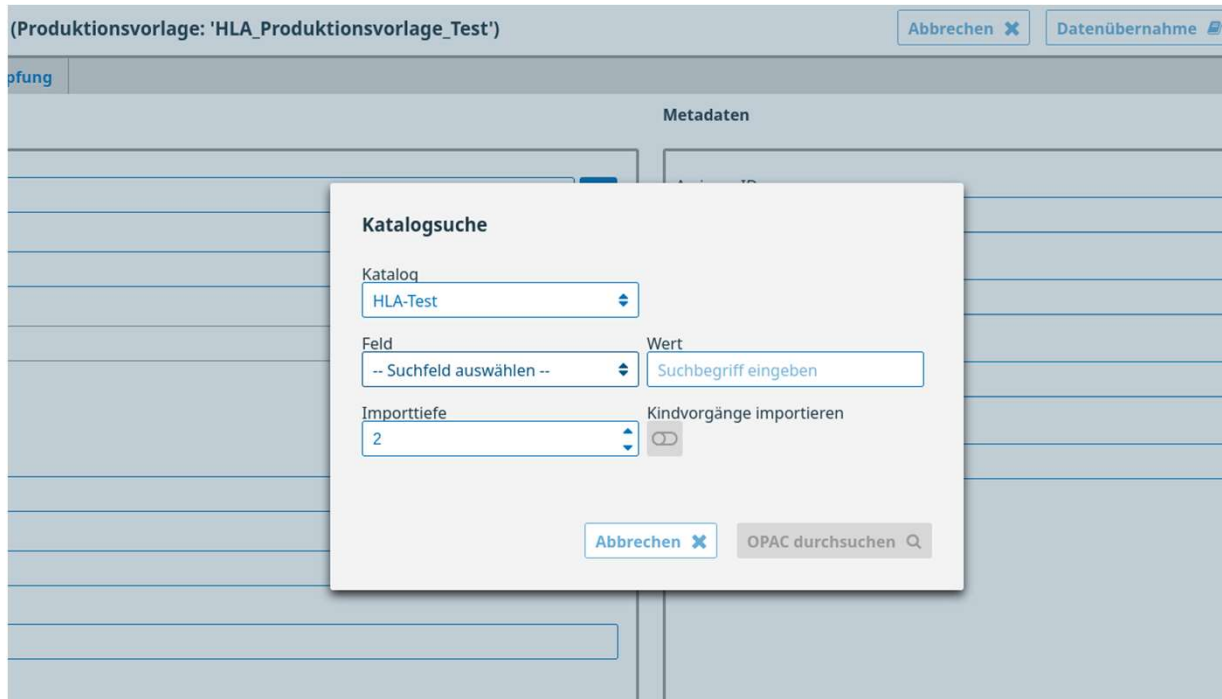
Aufbau einer provisorischen Schnittstelle zum Metadatenimport

Jörg Bieszczak (HRZ Uni Marburg)
Max Gerlings (HRZ Uni Marburg)
Nils Reichert (Hessisches Landesarchiv)

Berlin, den 07.11.2023

Datenimport

- Der Workflow eines Vorgangs beginnt **unbedingt** mit einem Import der Metadaten (aus einem „Katalogsystem“)
- Anhand eines **Identifiers** wird ein Datensatz (ggf. mit Eltern) per **Schnittstelle** geladen
- Alternativ kann per Massenimport eine Liste von Identifiern verarbeitet werden.
- Die momentan verwendete „Datenübernahme“ (manueller Upload) kann mehrere **Hierarchieebenen nur schwerlich abbilden**



(Produktionsvorlage: 'HLA_Produktionsvorlage_Test')

Abbrechen ✕ Datenübernahme

Metadaten

Katalogsuche

Katalog
HLA-Test

Feld Wert
-- Suchfeld auswählen -- Suchbegriff eingeben

Importtiefe Kindvorgänge importieren
2

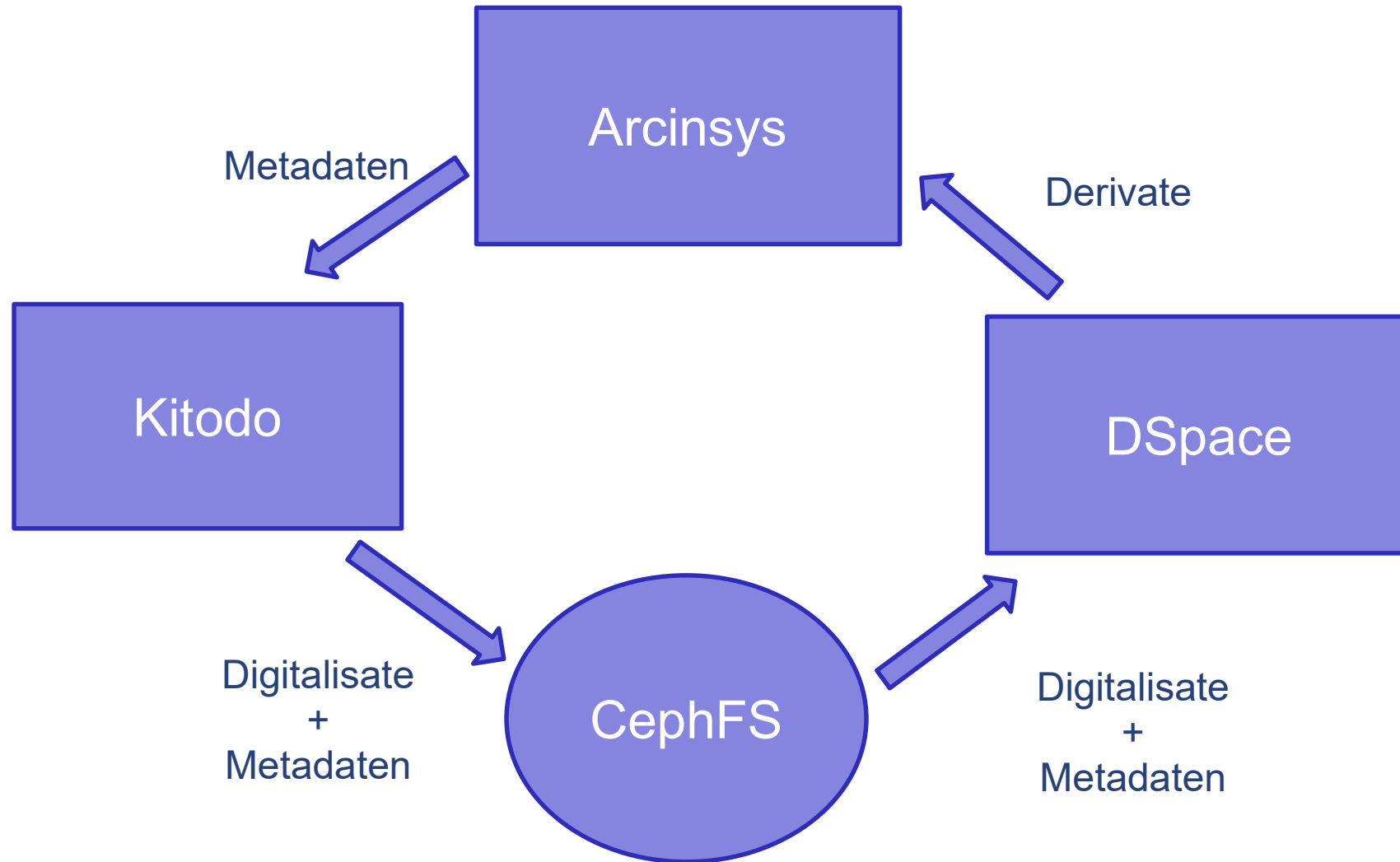
Abbrechen ✕ OPAC durchsuchen 🔍

Status Dateningest (Testphase)

- Verarbeitung von EAD-XML durch XSLT-Transformationen
 - Bislang nur einzelne Stücke, keine ganzen Bestände, kein Massenimport
- Zentrale Workflow-Funktion von Kitodo zur Verarbeitung von „Vorgängen“ im Aufbau
 - Potentiell Steuerung von Prozessen im Repo (wenn dieses entsprechende Schnittstellen/Rückgabewerte liefert) möglich
 - Möglichkeit zur automatischen Initiierung von Vorgängen/Workflows (z.B.durch Arcinsys) noch zu klären
 - Nach der Verarbeitung von Kitodo (strukturelle Metadaten, Erstellung Derivate) können Derivate z.B. an Arcinsys ausgeliefert werden.

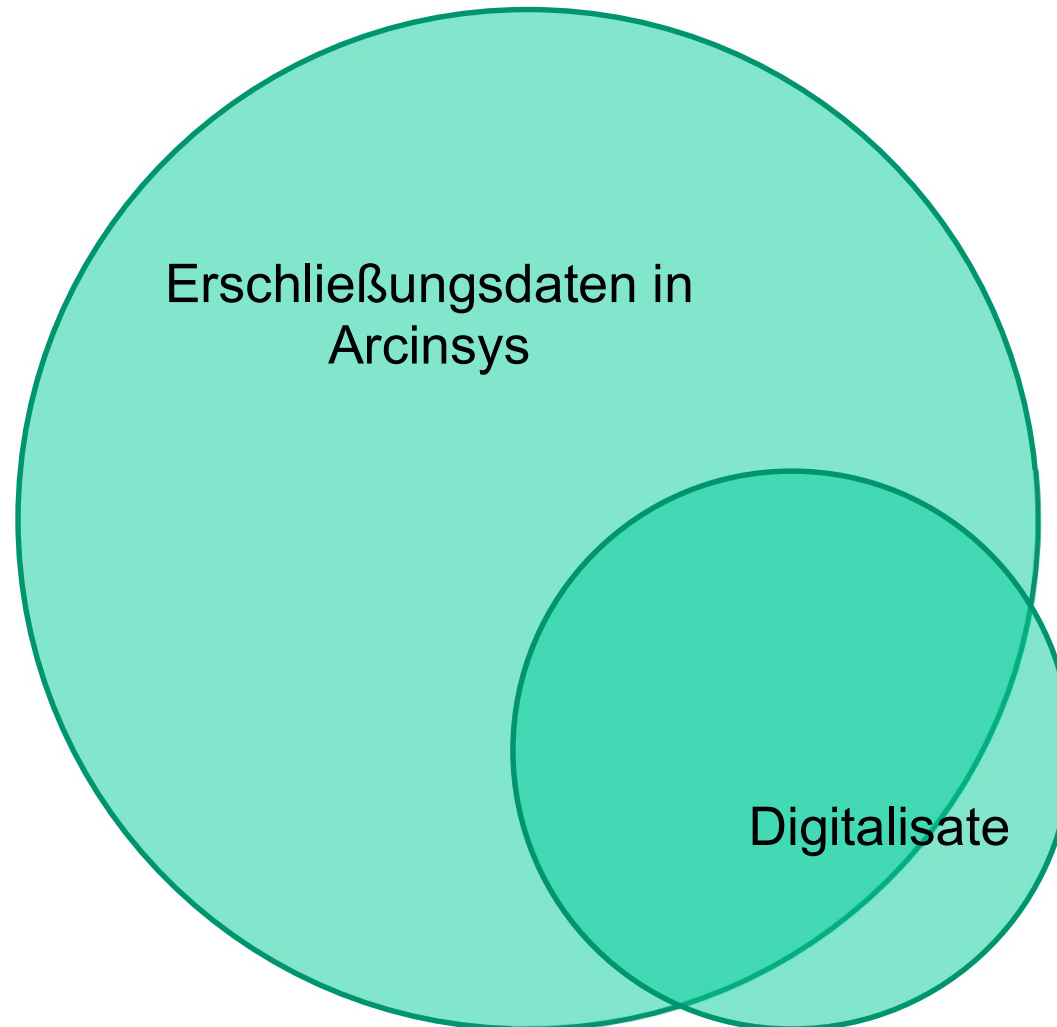


Circle of Life





Problem: Nicht übereinstimmende Datenmengen





RobinSerwe, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

Teil 1: Austausch zwischen Datensystemen


Ziel: Massenhaft die gewünschten u. benötigten Metadaten in
Kitodo aufnehmen

Pseudo-API(s) zum Datenimport mit BaseX

- Automatischer Export aller (öffentlichen) Erschließungsdaten aus Arcinsys
 - per cURL
 - über Exportfunktion der Arcinsys-Weboberfläche
 - in EAD-XML (EAD-DDB)
- Konfigurierbare REST-API

Export-Typ

Findbuch **EAD-XML** CSV Arcinsys-XML

Geschützte Elemente (verborgene Datensätze  und interne Notizen) mit exportieren

Keine Nur interne Notizen Nur verborgene Datensätze Beide

Digitalisate mit exportieren

Nein Ja

Export starten



Metadaten

Katalogsuche

Importkonfiguration

BaseX: (Arcinsys-)ID

BaseX: (Arcinsys-)ID

BaseX: Arcinsys-Signatur (Trennzeichen: ~)

BaseX: Pfad-Signatur

BaseX: Pseudo

BaseX: Pseudo-Multi

Abbrechen

Pseudo-API(s) zum Datenimport mit BaseX Teil 2

Bash-Script: Arcinsys2BaseX.sh

```
arcinsys_download_get_JOBID () {  
    echo "Start getting JOBID for ${1}..." 1>&3  
  
    START_EXPORT_RESULT=$(  
        curl --silent --show-error --fail \  
        -H "${JSESSION_COOKIE}" \  
        "${BASEURL_ARCINSYS}/system/ajaxStartXmlExport.action?itemid=${1}&visValue=no&exportDigitalisate=no&exportTyp=${2}"  
    )  
  
    if [[ "${START_EXPORT_RESULT}" != "" ]]; then  
        JOBID=$(echo "${START_EXPORT_RESULT}" | jq .job.id)  
        echo "Finished getting JOBID ${1}" 1>&3  
    else  
        error_exit "curl result is empty, no response from arcinsys! Session still alive?"  
    fi  
  
    echo "${JOBID}"  
}
```


TAXI-Format

- Token-Index in BaseX möglich
- Abfragegeschwindigkeit um Faktor 50 bis 300 erhöht!
- Anpassung der notwendigen Datenfelder auf wesentlich schlankere und effizientere Struktur. (XML-Lehrbuch ;-)

TAXI

Call it:

Tiny Archive XML Interchange Format

Tiny Archive eXchange Interface

Trivial Archive XML Interchange Format

Trivial Archive eXchange Interface

or "something, that converts supercomplicated archival transport xml-stuff into proper readable and structured xml, like real people do."

```

<?xml version="1.0" encoding="UTF-8"?>
<ead ...
  <eadheader ...
    <eadid mainagencycode="">b7205</eadid>
    ...
  </eadheader>
  <archdesc level="collection" type="Findbuch">
    <did>
      ...
    </did>
    ...
    <dsc>
      <c level="collection" id="b7205">
        <did>
          ...
        </did>
        ...
        <c level="class" id="g257657">
          <did>
            ...
          </did>
          <c level="file" id="v2256588">
            <did>
              <unitid>43</unitid>
              <unittitle>Liederbuch von
...</unittitle>
              <physdesc>
                <genreform>Sachakte</genreform>
              </physdesc>
              <unitdate normal="1920-01-01/1920-12-31">
                1920
              </unitdate>
            </c>
          </c>
        </dsc>
      </c>
    </archdesc>
  </ead>

```

```

<?xml version="1.0" encoding="utf-8"?>
<taxi ...
  <object type="genericitem" id="v2256588"
  fsig="adjb/ch_2/43" asig="AdJB~CH 2~43">
    <item>
      <id>v2256588</id>
      <signature>
        <arcinsys>
          <current>43</current>
          <previous>CH 1043</previous>
        </arcinsys>
        <path>
          <current>43</current>
          <previous>ch_1043</previous>
        </path>
      </signature>
      <title>Liederbuch von ...</title>
    </item>
    <document_type>ThematicFile</document_type>
    <genre>Sachakte</genre>
    <inclusive_dates>
      <text>1920</text>
      <begin>1920-01-01</begin>
      <end>1920-12-31</end>
      <period>1920-01-01/1920-12-31</period>
    </inclusive_dates>
    <content>
      <text>32 Blatt</text>
    </content>
  </object>
  <fond>
    ...
  </fond>
  <archive>
    ...
  </archive>

```





Hierarchischer Massenimport

- Regelsätze
- Identifier
- Überordnung

```
<key id="ArcinsysID" use="recordIdentifier">
  <label>Arcinsys ID</label>
</key>
<key id="stockID" use="higherLevelIdentifier">
  <label>Stock-ID</label>
  <label lang="de">Bestands ID</label>
```

```
-<taxi>
  -<object type="genericitem" fsig="hstam/bestand/stueck">
    -<item>
      <document_type>Unknown</document_type>
      -<signature>
        -<path>
          <current>hstam/bestand/stueck</current>
        </path>
      </signature>
    </item>
  </object>
</taxi>
```

```
<xsl:template match="taxi:object/taxi:fond/taxi:id">
  <kitodo:metadata name="stockID">
    <xsl:value-of select="normalize-space()"/>
  </kitodo:metadata>
</xsl:template>

<xsl:template match="taxi:object[@id]">
  <kitodo:metadata name="ArcinsysID">
    <xsl:value-of select="@id"/>
  </kitodo:metadata>

  <xsl:apply-templates select="@*|node()"/>
</xsl:template>
```

Hierarchischer Massenimport + Pseudoschnittstellen

KITODO PRODUCTION Suche Massenimport

Massenimport (Produktionsvorlage: 'HLA_Produktionsvorlage_SiFi')

Massenimport

Katalog
BaseX: (Arcinsys-)ID + Auswählen

CSV-Datei-Upload + Auswählen

Datensätze
CSV-Spaltentrennzeich :

ID	
b5212	
v2880367	
v2034057	
v907494	
v1822899	
v4914969	
v2495303	
v5438367	
v2034060	
v2495291	

KITODO PRODUCTION Suche Massenimport

Massenimport (Produktionsvorlage: 'HLA_Produktionsvorlage_SiFi')

Massenimport

Katalog
BaseX: Pseudo + Auswählen

CSV-Datei-Upload + Auswählen

Datensätze
CSV-Spaltentrennzeich :

ID	
hhstaw/220/971	
hstam/protokolle/ii_grebenstein_29_bd_6	
hstam/protokolle/ii_grebenstein_19b	
hstam/protokolle/ii_grebenstein_30_bd_1	
hstam/495/p_ii_16904	
hstam/495/p_ii_17224	
hstam/495/p_ii_16972	
hstam/495/p_ii_16504	
hstam/495/p_ii_16309	

KITODO PRODUCTION Suche Massenimport

Massenimport (Produktionsvorlage: 'HLA_Produktionsvorlage_SiFi')

Massenimport

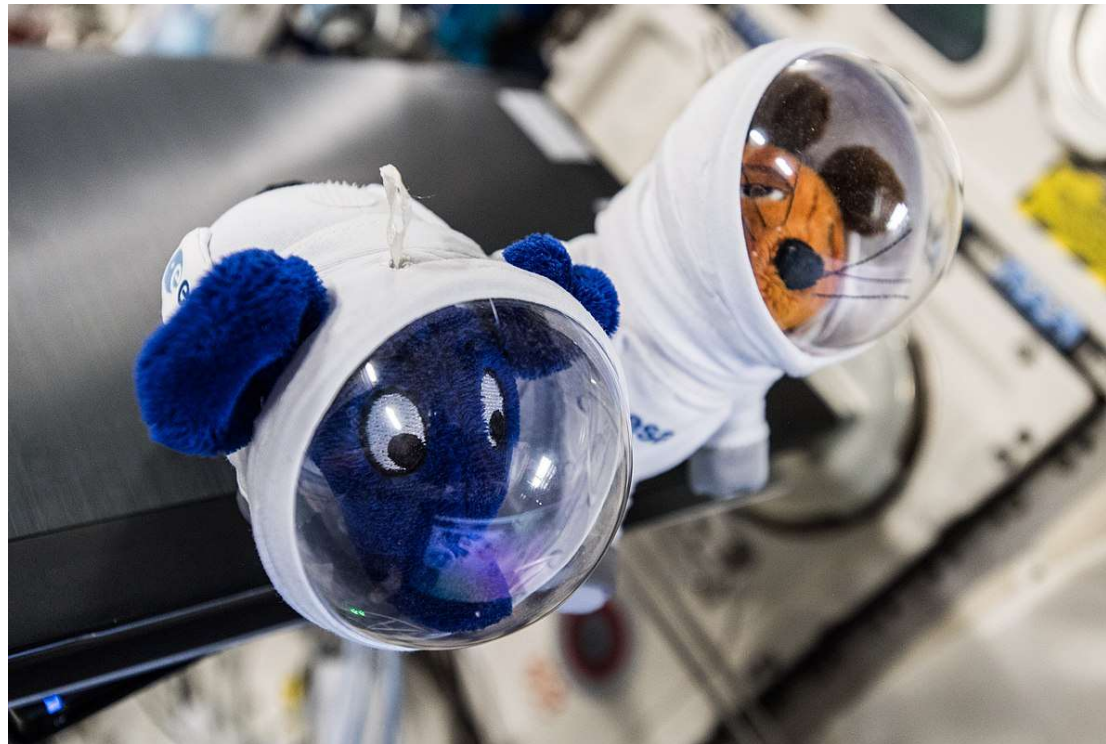
Katalog
BaseX: Pseudo-Multi + Auswählen

CSV-Datei-Upload + Auswählen

Datensätze
CSV-Spaltentrennzeichen :

ID	
hhstaw/220/871__v907493;v3685392;	
hstam/protokolle/ii_grebenstein_23__v520288;v2510413;v3886167;v5279574;	

+ Auswählen



Alexander Gerst, CC BY-SA 2.0 <<https://creativecommons.org/licenses/by-sa/2.0/>>, via Wikimedia Commons

Teil 2: Prozesse starten

Ziel: Abbildung des einfachsten und regelmäßigsten bisherigen Workflows

(Landesnutzen aus Bundessicherungsverfilmung)

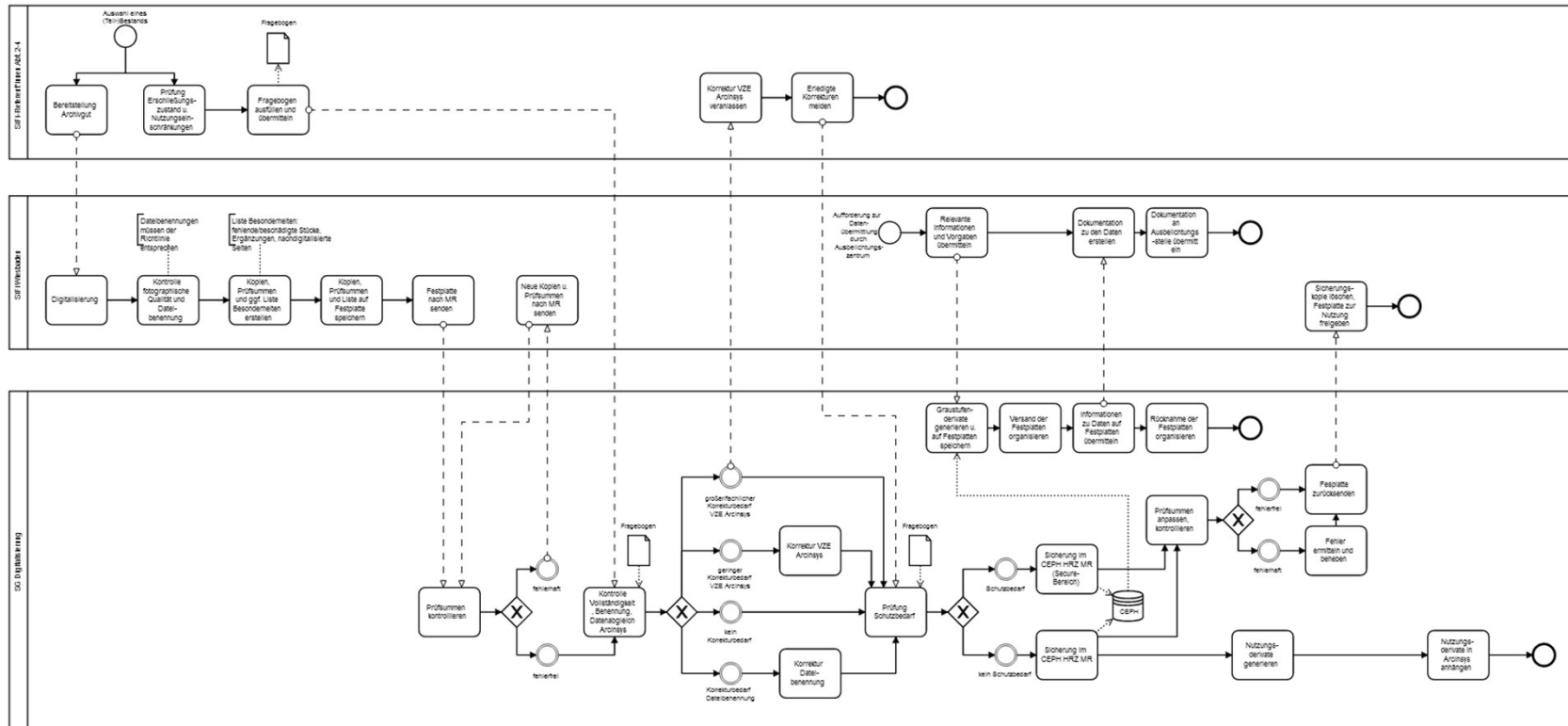
Workflows („Step by Step“)

- Workflows werden oft projektweise definiert
 - bisweilen auch erst nach der Durchführung der Digitalisierung
- Wenige dauerhaft und regelmäßig laufende Workflows
 - Bundessicherungsverfilmung (ca. 2-monatlich, relativ wenig Überraschungen)
 - Reprographieaufträge für Nutzer*innen (halbjährlich, Umfang vorher unbekannt)
 - Längerfristige Kooperationsprojekte

Ein erster Workflow

- Notwendige Anpassung der Benennungsstruktur (Zuordnung zu Metadaten)
- Abgleich Schutzbedarf und differenzierte Behandlung
- Korrekturmeldungen an Erschließung
- Vorbereitung Langzeitspeicherung
- Derivate erzeugen und veröffentlichen
- Festplatten löschen
- An Ausbelichtungsstelle versenden

Der Workflow – wie er war

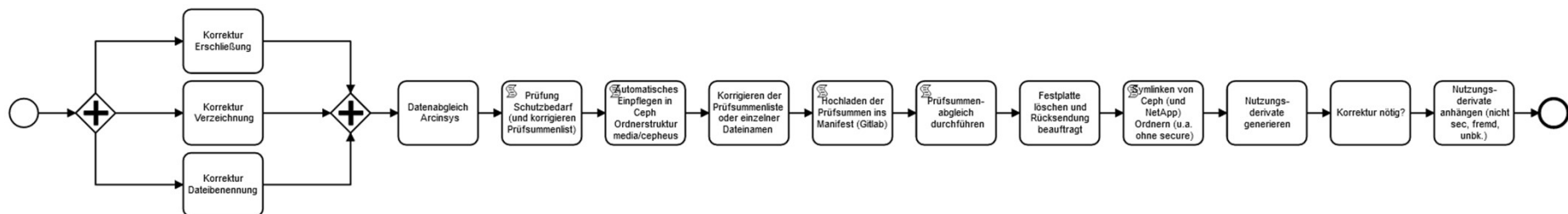


Verzweigungen dienen der Abbildung von:

- Korrektur- und Schutzfällen
- von einander unabhängige weiteren Verarbeitungen

Der Workflow – wie er ist

- Keine Verzweigungen möglich (!!!)
(siehe <https://github.com/kitodo/kitodo-production/issues/5630>)
- Kein initialer Abgleich als Teil des Workflows
- Gleicher Workflow für alle Tektonikebenen im gleichen Projekt





Pre-Ingest Ingest-Workflow

- Anlegen von (sinnvollen) Vorgängen setzt Datenabgleich voraus
- Automatischer Prozess zum Datenabgleich findet **vor** dem Workflow statt
- z.T. manuelle Korrekturen **vor** dem Pre-Ingest nötig
- **vor** eventuellen Korrekturen Upload in Ingest-Speicher

```

-- all_match.csv
-- all_multi_match.csv
-- all_no_match.csv
-- hhstaw
  |-- 220
  |   |-- matched.csv
  |   |-- more_than_one_match_hhstaw___220___871.txt
  |   |-- no_match.csv
  -- hstad
     |-- g_27_darmstadt
     |   |-- matched.csv
     |-- g_55
     |   |-- matched.csv
     |   |-- no_match.csv
  -- hstam
     |-- 495
     |   |-- no_match.csv
     |-- protokolle
     |   |-- matched.csv
     |   |-- more_than_one_match_hstam___protokolle___ii_grebenstein_23.txt
     |   |-- no_match.csv
  -- stadt_kb
     |-- historische_akten
     |   |-- no_match.csv
10 directories, 14 files

```

```

==> all_match.csv <==
ID
b5212
v2880367
v2034057
v907494

==> all_multi_match.csv <==
ID
hhstaw/220/871___v907493;v3685392;
hstam/protokolle/ii_grebenstein_23___v520288;v2510413;v3886167;v5279574;

==> all_no_match.csv <==
ID
hhstaw/220/971
hstam/protokolle/ii_grebenstein_29_bd._6
hstam/protokolle/ii_grebenstein_19b
hstam/protokolle/ii_grebenstein_30_bd._1

```

Pre-Ingest Ingest-Workflow Teil 2: Script

```

#extract archive from item
archive=$(echo "${item}" | awk -F / '{print $1}')
#extract fond from item
fond=$(echo "${item}" | awk -F / '{print $2}')
#create archive and fond folder
mkdir -p "${base_path_tmp}/${archive}/${fond}"
#check baseex for item by signature
arcid=$(curl --silent --globoff "${url_baseex}?${parameter_baseex}=${item}")
#convert output to only get ids without header and xml
xmlout=$(echo "${arcid}" | xmllint sel -t -v "/" 2>/dev/null)
#count the occurrence of the letter 'v' (every arcinsysid starts with v)
count=$(echo "${xmlout}" | grep -Fo v | wc -l)
#check if more than one arcinsysid
if [[ "${count}" -gt 1 ]]; then
    filename="${item//\\/_}"
    echo "More than one matching, tell this Arcinsys! | Signatur: ${item}"
    echo -n "${item}__" > "${base_path_tmp}/${archive}/${fond}/more_than_one_match_${filename}.txt"
    echo "${xmlout}" | tr " " "\n" >> "${base_path_tmp}/${archive}/${fond}/more_than_one_match_${filename}.txt"
    #replace newline by _ and add to all multi match file
    awk -v ORS=";" '1' "${base_path_tmp}/${archive}/${fond}/more_than_one_match_${filename}.txt" >> "${base_path_tmp}/all_multi_match.csv"
    #write newline to all multi match file
    echo "" >> "${base_path_tmp}/all_multi_match.csv"
#check if exactly one matching arcinsysid
elif [[ "${count}" -eq 1 ]]; then
    echo "Exact one matching! | Signatur: ${item} | ArcinsysID: ${xmlout}"
    echo "${xmlout}" >> "${base_path_tmp}/${archive}/${fond}/matched.csv"
#no matching arcinsysid

```

Pre-Ingest Ingest-Workflow: Perspektive

- Ein Massen-Reimport könnte dies ändern:
 1. Anlegen der Vorgänge aus den Pfaden zu den Digitalisaten
 2. Über Kitodo gesteuerter Abgleich mit Arcinsys
- ↳ Durch Reimport Anreichern der Metadaten nach dem Abgleich
 - hierbei wären insbesondere der Doctype und auch der Vorgangstitel wichtig

Metadaten aktualisieren (PPN1234567, Kalliope) ?

Metadatum	Bisheriger Wert	↔	Neuer Wert
Titel	Lorem ipsum	←	Lorem ipsum 2
Sprache	de	↔	en
Erscheinungsjahr	1900	←	1905
Erscheinungsort	Hamburg	→	Harburg

...



PantheraLeo1359531, CC BY 4.0 <<https://creativecommons.org/licenses/by/4.0/>>, via Wikimedia Commons

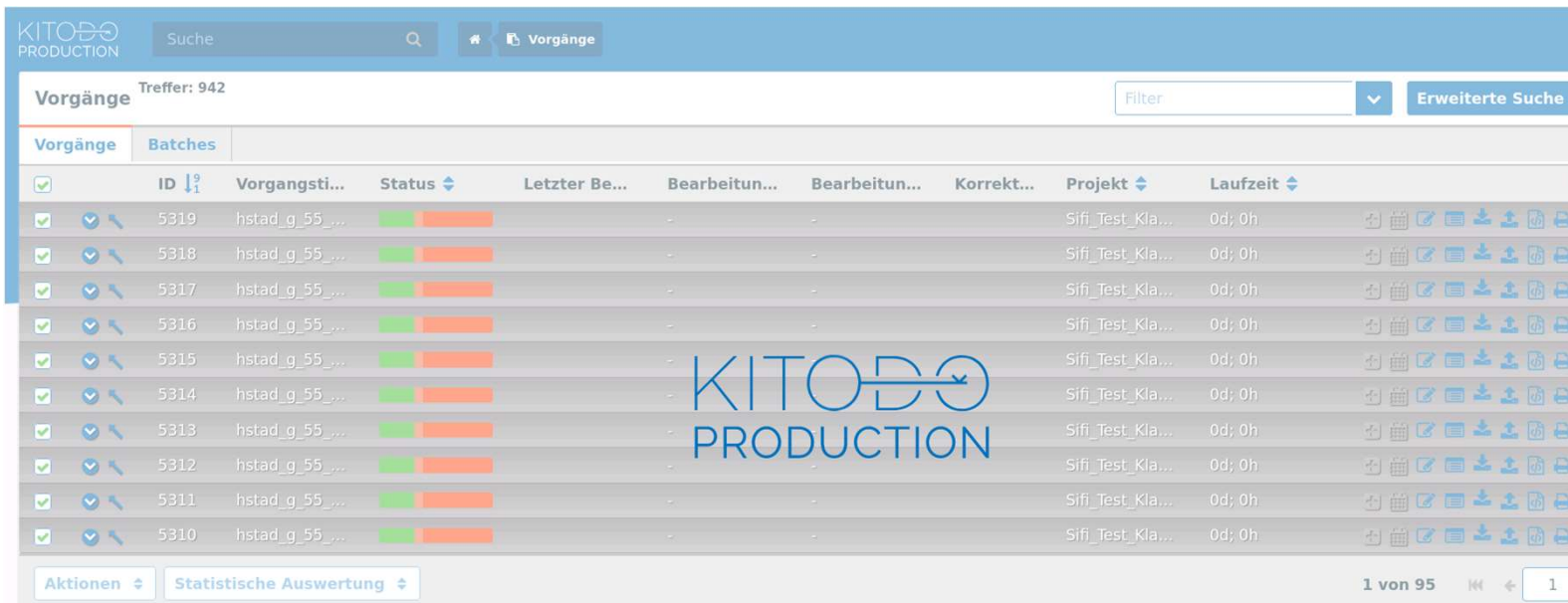
Teil 3: Prozesse durchführen

Ziel: Alle Workflow-Komponenten, die keiner menschlichen Entscheidung bedürfen, sollen automatisiert werden

Automatische massenhafte Datenverarbeitung

Performance:

- Import (CSV): circa 15min für ungefähr 1500 Vorgänge
- Schutzstatus prüfen und anschließendes Verschieben der Dateien vom Ingest in finale Struktur: ca. 1,5Std für ungefähr 1500 Vorgänge



Vorgänge Treffer: 942

Vorgänge	Batches	ID	Vorgangsti...	Status	Letzter Be...	Bearbeitun...	Bearbeitun...	Korrekt...	Projekt	Laufzeit
<input checked="" type="checkbox"/>	<input type="checkbox"/>	5319	hstad_g_55_...	<div style="width: 100%;"><div style="width: 100%;"></div></div>	-	-	-	-	Sifi_Test_Kla...	0d; 0h
<input checked="" type="checkbox"/>	<input type="checkbox"/>	5318	hstad_g_55_...	<div style="width: 100%;"><div style="width: 100%;"></div></div>	-	-	-	-	Sifi_Test_Kla...	0d; 0h
<input checked="" type="checkbox"/>	<input type="checkbox"/>	5317	hstad_g_55_...	<div style="width: 100%;"><div style="width: 100%;"></div></div>	-	-	-	-	Sifi_Test_Kla...	0d; 0h
<input checked="" type="checkbox"/>	<input type="checkbox"/>	5316	hstad_g_55_...	<div style="width: 100%;"><div style="width: 100%;"></div></div>	-	-	-	-	Sifi_Test_Kla...	0d; 0h
<input checked="" type="checkbox"/>	<input type="checkbox"/>	5315	hstad_g_55_...	<div style="width: 100%;"><div style="width: 100%;"></div></div>	-	-	-	-	Sifi_Test_Kla...	0d; 0h
<input checked="" type="checkbox"/>	<input type="checkbox"/>	5314	hstad_g_55_...	<div style="width: 100%;"><div style="width: 100%;"></div></div>	-	-	-	-	Sifi_Test_Kla...	0d; 0h
<input checked="" type="checkbox"/>	<input type="checkbox"/>	5313	hstad_g_55_...	<div style="width: 100%;"><div style="width: 100%;"></div></div>	-	-	-	-	Sifi_Test_Kla...	0d; 0h
<input checked="" type="checkbox"/>	<input type="checkbox"/>	5312	hstad_g_55_...	<div style="width: 100%;"><div style="width: 100%;"></div></div>	-	-	-	-	Sifi_Test_Kla...	0d; 0h
<input checked="" type="checkbox"/>	<input type="checkbox"/>	5311	hstad_g_55_...	<div style="width: 100%;"><div style="width: 100%;"></div></div>	-	-	-	-	Sifi_Test_Kla...	0d; 0h
<input checked="" type="checkbox"/>	<input type="checkbox"/>	5310	hstad_g_55_...	<div style="width: 100%;"><div style="width: 100%;"></div></div>	-	-	-	-	Sifi_Test_Kla...	0d; 0h

Aktionen Statistische Auswertung 1 von 95

Automatische massenhafte Datenverarbeitung

Herausforderungen:

- Zu viel Logging führt zu Deadlock und fehlerhaften Prozessen
- Erster Workflowschritt kann nicht automatisch sein (siehe <https://github.com/kitodo/kitodo-production/issues/3672>)
- Bearbeitung von Vorgängen ist möglich, noch während automatische Verarbeitung läuft
- Verändern des Bearbeitungsstatus eines Vorgangs wird direkt übernommen, klicken auf „Speichern“ nicht erforderlich → mögliche Fehlerquelle
- Massenhaftes Löschen von Vorgängen bisher nicht möglich (siehe <https://github.com/kitodo/kitodo-production/issues/5342>)



Rigorius, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

**VIELEN DANK FÜR DIE
AUFMERKSAMKEIT!**