

# BeLab Workshop

Karlsruhe - 4. November 2011

## Implementierung des BeLab-Systems

**Referent:** Jan Potthoff

**Projektleiter:**

Prof. Dr. Bernhard Neumair (SCC)

Prof. Dr. Alexander Roßnagel (Uni Kassel)

Prof. Dr. Siegfried Hackel (PTB)

STEINBUCH CENTRE FOR COMPUTING - SCC



# Gliederung

- **Das BeLab-Projekt**
  - Motivation und Ziele
  - Verlauf (Versionshistorie)
  
- **Implementierung: BeLab-System/Service**
  - Schnittstelle und Funktionen des BeLab-Services
  - Datenmodell und die interne Architektur des BeLab-Systems
  - Module des BeLab-Systems

# Beweissicheres elektr. Laborbuch (BeLab)

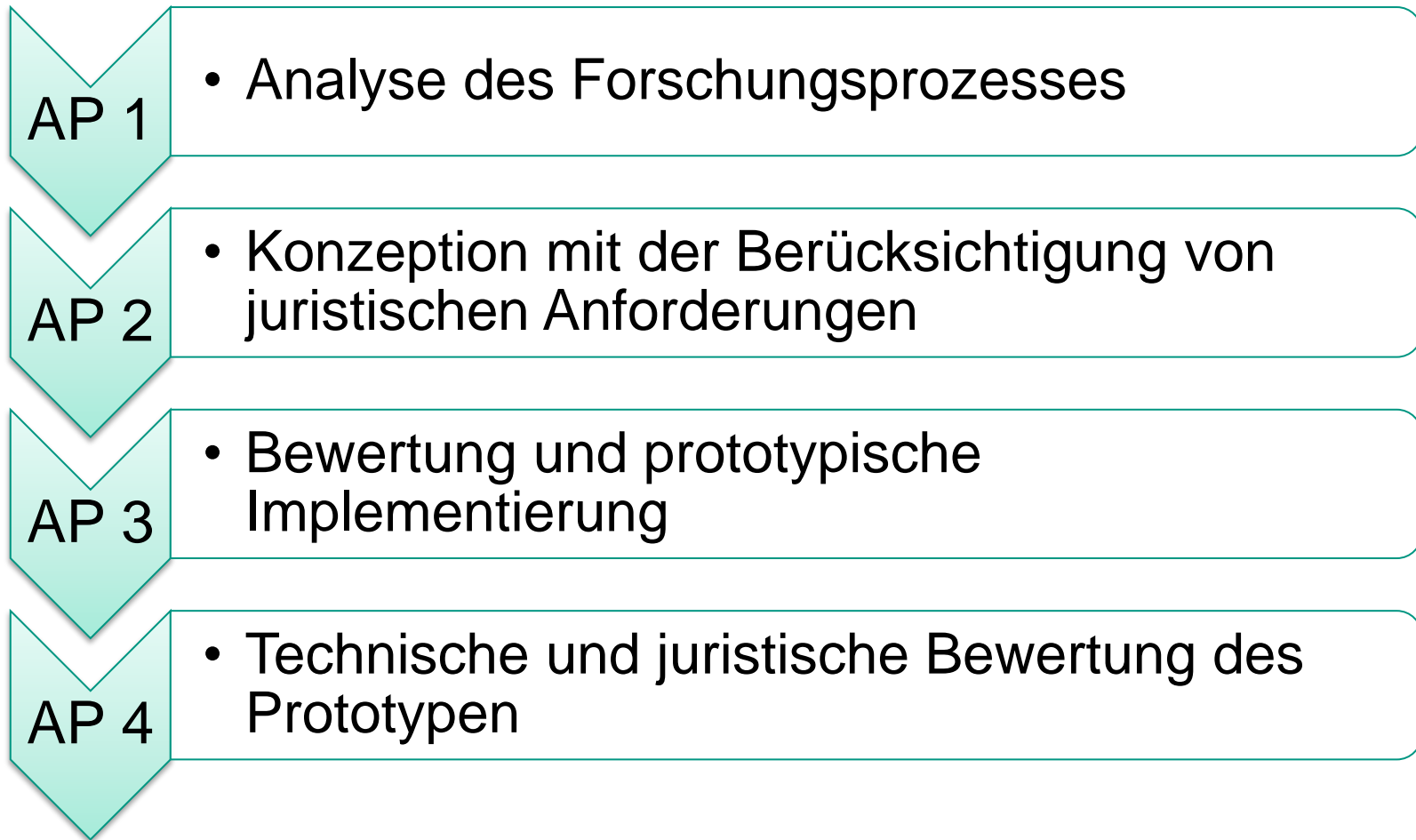
## ■ Daten im Forschungsprozess

- Häufig elektronisch (Tendenz steigend)
- Übliche Verwaltung/Dokumentation
  - Papiergebundenes Laburbuch
  - Trend: Elektronische Verwaltung
- Vorgaben
  - Nachvollziehbare Dokumentation
  - Archivierung der Forschungsdaten (Bsp.: DFG mind. 10 Jahre)

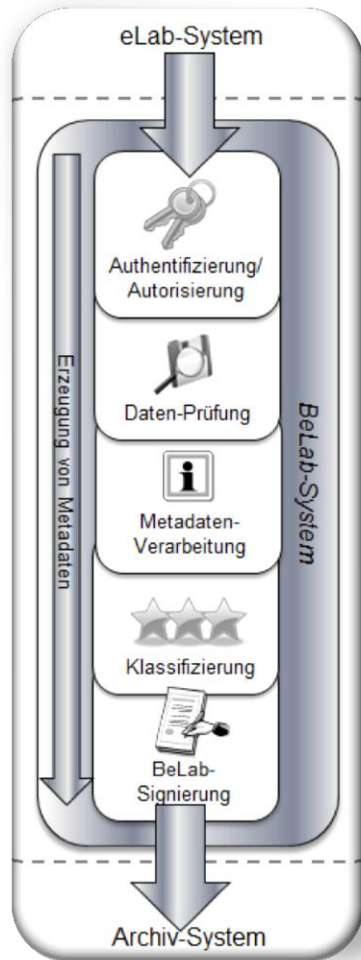
## ■ BeLab-Projekt

- Ziel: Vergleichbare Verbindlichkeit zu papiergebundenen Laborbüchern
- Projektpartner:
  - Universität Kassel
  - Physikalisch-Technische Bundesanstalt (PTB)
  - Karlsruher Institut für Technologie (KIT)
- Start: 01.02.2010

# Arbeitsschritte des BeLab-Projekts



# Konzeption des BeLab-Systems



- Generische Schnittstelle zu eLabs
- Authentifizierung, Autorisierung
- Datenübergabe/-verwaltung
  - Angabe von Metadaten
  - Eindeutige Identifikation der Daten
- Datenüberprüfung
  - Dateiformat, Signatur
  - Metadaten, Datenvollständigkeit
- Klassifizierung
  - LZA-Tauglichkeit
  - Signaturverfahren
  - Beweiskraft
- Signierung und Archivierung

# Versionshistorie (I)

- Version 0.0.1 (01.03.2011):
  - Daten können an den BeLab-Service übergeben werden
  - Passwort-basierte Authentifizierung
  - Rückgabewert: „Erfolgreicher Upload“
- Version 0.0.2 (13.04.2011):
  - Strukturierung des BeLab-internen Ablaufs
  - Anpassungen der Schnittstelle für PHP und Python
- Version 0.0.3 (13.05.2011):
  - Passwort- und zertifikatsbasierte Authentifizierung
  - Autorisierungsfunktion
  - Daten werden auf Server gespeichert + DatenID
  - Abschaffung der BeLab-Exception (stattdessen „immer“ String)
  - Verwendbare Funktion: archiveData

## Versionshistorie (II)

- Version 0.0.4 (20.07.2011):
  - Daten müssen nach der UOF-Definition übergeben werden
  - Erste Datenüberprüfungsmethode (signierte Bilder)
  - Logging des Verarbeitungsprozesses
  - Metadaten können abgerufen werden (retrieveMetadata)
  - Test-Client als Demonstrator des BeLab-Services
  
- Aktuellste Version: 0.1.0 (24.11.2011)

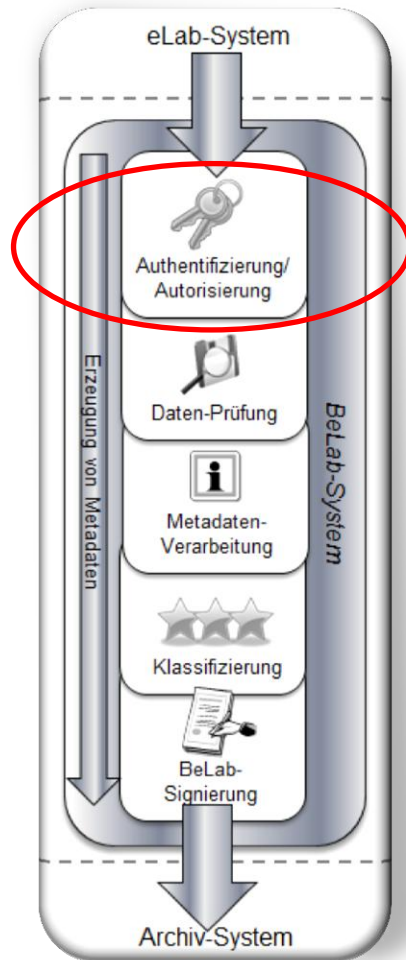
[https://scc-belab.scc.kit.edu:8443/BeLab\\_V1/services/BeLabPBService?wsdl](https://scc-belab.scc.kit.edu:8443/BeLab_V1/services/BeLabPBService?wsdl) bzw.  
[https://scc-belab.scc.kit.edu:8443/BeLab\\_V1/services/BeLabCBService?wsdl](https://scc-belab.scc.kit.edu:8443/BeLab_V1/services/BeLabCBService?wsdl)

# Gliederung

- Das BeLab-Projekt
  - Motivation und Ziele
  - Verlauf (Versionshistorie)
- Implementierung: BeLab-System/Service
  - Schnittstelle und Funktionen des BeLab-Services
  - Datenmodell und die interne Architektur des BeLab-Systems
  - Module des BeLab-Systems

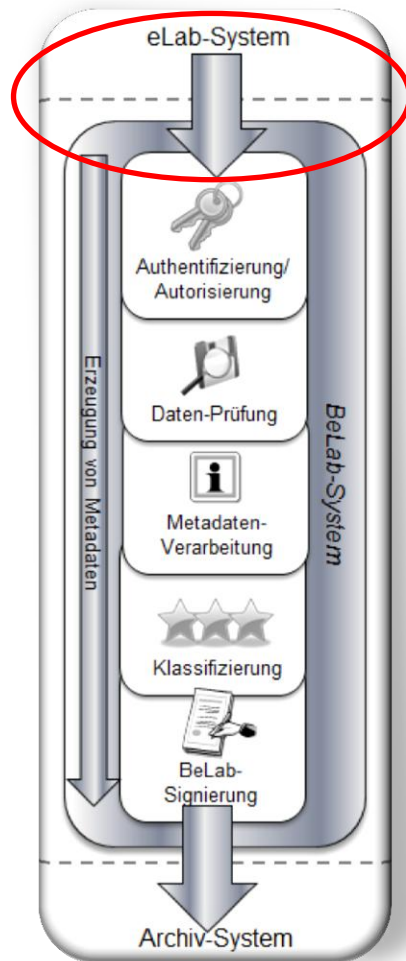


# Schnittstelle: Elektronisches Laborbuch



- **Web Service**
  - Framework: Axis2 (Apache Software Foundation), Version 1.3
  - Protokoll: SOAP
- **Authentifizierung**
  - Hypertext Transfer Protocol Secure (HTTPS)
  - Client-Authentifizierung (Rampart 1.3):
    - Zertifikatsbasiert (DFN-Zertifikat, Test-Zertifikat)
    - Passwort-basiert
- **Autorisierung: XML-Definition**
- **Benutzerinterface**
  - Systemhinweise
  - Administration

# Web Service: Datenverwaltung

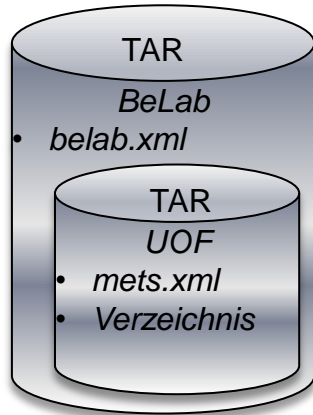


- Datenimport
  - Formatüberprüfung (UOF)
  - BeLab-Prozess
  - Archivierung
- Datenupdate
  - Formatüberprüfung
  - BeLab-Prozess
  - Versionierung
- Datenexport
- Stornieren von Daten
- (Suchfunktion)

# Gliederung

- Das BeLab-Projekt
  - Motivation und Ziele
  - Verlauf (Versionshistorie)
- Implementierung: BeLab-System/Service
  - Schnittstelle und Funktionen des BeLab-Services
  - Datenmodell und die interne Architektur des BeLab-Systems
  - Module des BeLab-Systems

# Daten- und Metadatenmodell



```
<mets>
```

```
<metsHdr>
```

```
  <agent>...</agent>
```

```
</metsHdr>
```

```
<dmdSec id=001>...</dmdSec>
```

```
<amdSec id=000>...</amdSec>
```

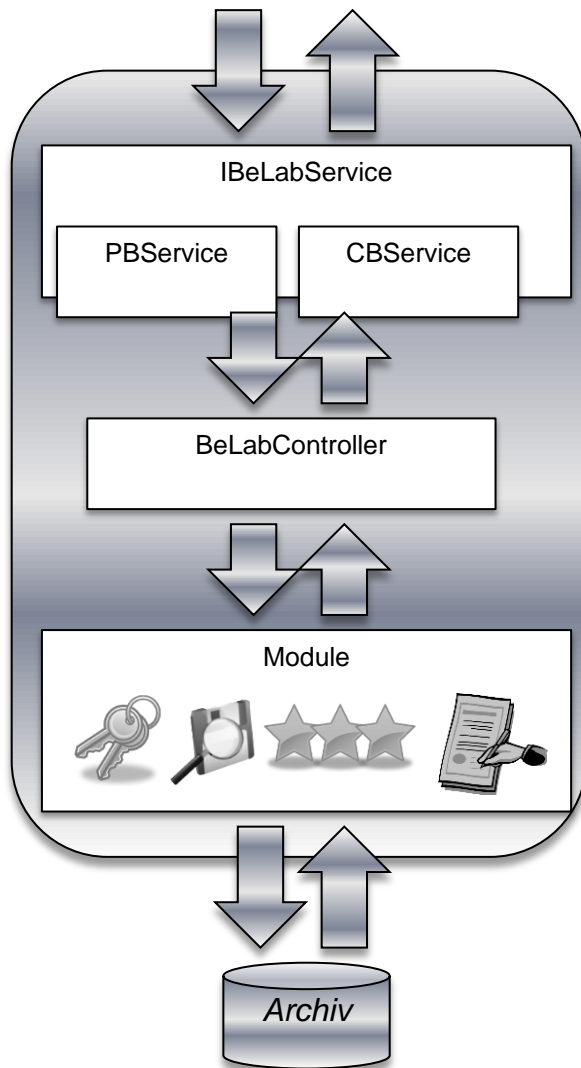
```
<fileSec>...</fileSec>
```

```
...
```

```
</mets>
```

- Universal Object Format (UOF)
  - Projekt: kopal
  - TAR- oder ZIP-Archiv
  - root-Verzeichnis + mets.xml
- BeLab-Datenformat
  - UOF-Archiv
  - belab.xml
- Datenformate
  - XML
  - Dateien
  - Hashwerte
- Angabe von Metadaten
- Formatüberprüfung beim Datenimport

# BeLab: Interne Architektur

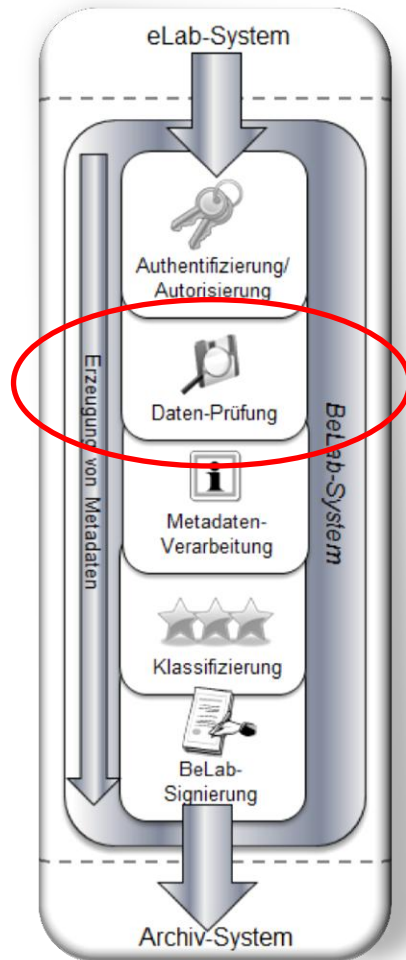


- Nach der Formatüberprüfung
  - Übergabe der Daten an den Controller
  - Aufbereitung der Daten
    - Datahandler: Dateizugriff (Apache Commons VFS)
    - Metadatahandler: Metadatenanalyse, -erzeugung (METS Java Toolkit)
  - Aufruf der einzelnen Module
- Umsetzung der Module
  - Modul-basiert
  - Modul-Controller übernimmt Ausführung
  - Rückgabe des Ergebnisses basiert auf BeLab-Modell

# Gliederung

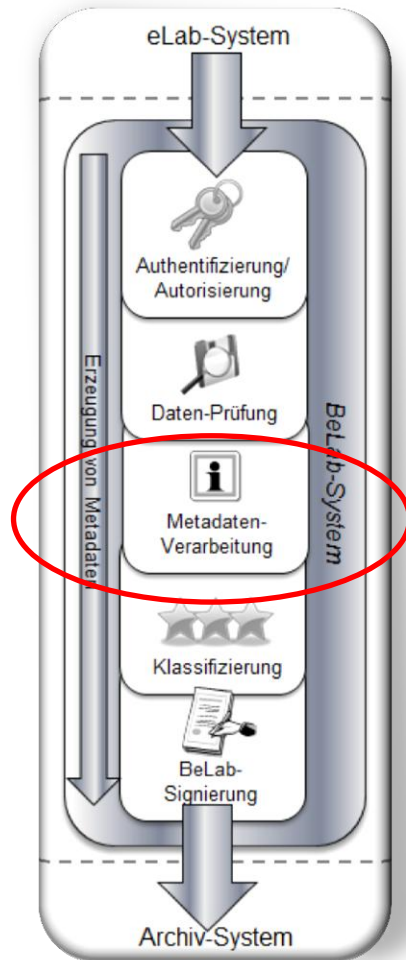
- Das BeLab-Projekt
  - Motivation und Ziele
  - Verlauf (Versionshistorie)
- Implementierung: BeLab-System/Service
  - Schnittstelle und Funktionen des BeLab-Services
  - Datenmodell und die interne Architektur des BeLab-Systems
  - Module des BeLab-Systems

# Datenprüfung



- Überprüfung
  - Integrität und Authentizität
  - Beweiswert
  - LZA-Tauglichkeit
- Ergebnis
  - Darlegung der „Beschaffenheit“ der Daten
  - Rückmeldung über mögliche Risiken
- Beispiele
  - Checksumme
  - Bilddatei (Kappa)
- Aufzeichnung der Ergebnisse

# Metadaten (Nachvollziehbarkeit)



## ■ Metadaten

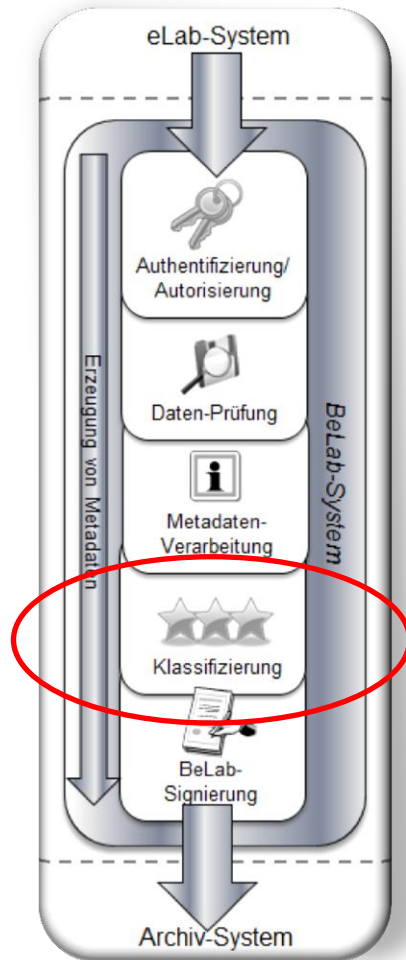
- Analyse der übergebenen Daten
- Verwaltung der Daten im Archiv
- Dokumentation des BeLab-Prozesses
  - Erfolgreiches Anmelden
  - Ergebnis der Formatüberprüfung und Datenprüfung
  - Ergebnis der Klassifizierung
  - Weitere Prozessschritte

## ■ Zusätzliches Logging: log4j

- Fehleranalyse
- Detaillierter Prozessablauf
- Log-Datei mit Prozess-ID

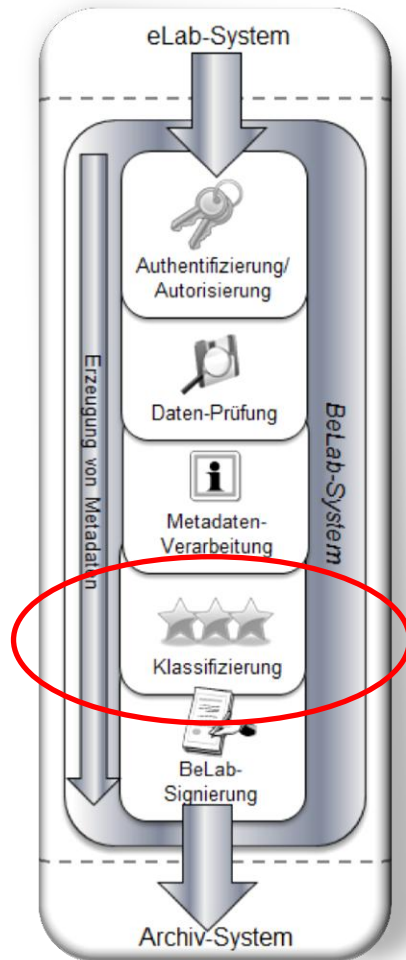


# Klassifizierung (I)



- Auswertung der Datenprüfung
- **Signatur**
  - B1 - keine
  - B2 - fortgeschritten
  - B3 - qualifiziert
  - B4 - qualifiziert+ (inkl. Zeitstempel)
  - B5 - akkreditiert (akkreditierter Zertifikatsdiensteanbieter)
  - B6 - akkreditiert+ (inkl. Zeitstempel)

# Klassifizierung (II)



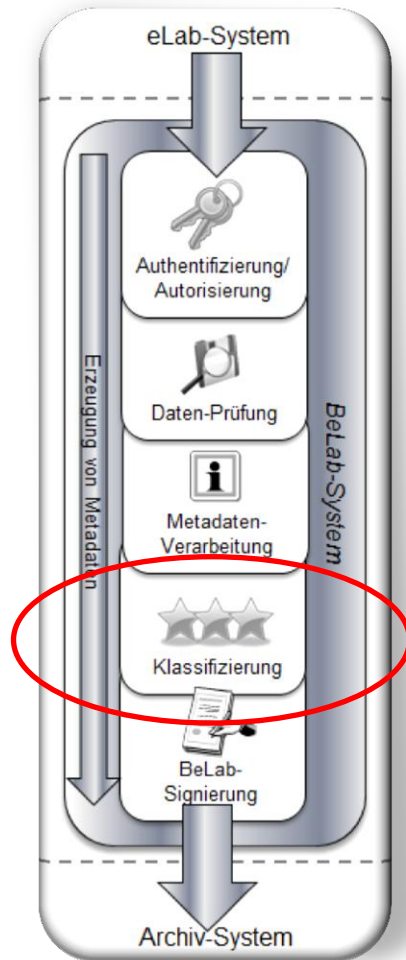
## ■ Dateiintegrität

- S1 - ungesicherte Datenerzeugung
- S2 - gesicherte Datenerzeugung

## ■ Datenformat

- L1 - ungeeignet
- L2 - geeignet (Bsp.: PDF, TIFF, DOC)
- L3 - empfohlen (Bsp.: ASCII, XML, PDF/A)

# Klassifizierung (III)



## ■ Beispiel: Bilddatei (Kappa)

- Ergebnis: Signatur gültig / nicht gültig
- Bei gültiger Signatur:
  - LZA-Tauglichkeit: L1 (ungeeignet)
  - Beweissicherheit: (hier) B6
  - Sicherung der Integrität: S2
  - Klassifizierung: **B6, L1, S2**

## ■ Gesamtbewertung:

- B6, L1, S2
- U , L2, S2
- B6, L1, S2
- Erschleichen einer hohen Klassifizierungen nicht möglich



# Vielen Dank für Ihre Aufmerksamkeit!

Karlsruher Institut für Technologie (KIT)  
Steinbuch Centre for Computing (SCC)

Dipl.-Inform. Jan Potthoff

Hermann-von-Helmholtz-Platz 1  
76344 Eggenstein-Leopoldshafen

Tel.: +49 721 / 608-25666  
E-Mail: [jan.potthoff@kit.edu](mailto:jan.potthoff@kit.edu)