



# WissGrid

## Dokumentation



Bundesministerium  
für Bildung  
und Forschung

### Arbeitspaket 3 - Langzeitarchivierung von Forschungsdaten

## Integration der Systeme Fedora Commons und iRODS<sup>1</sup>

Autoren	Arbeitspaket 3 - Langzeitarchivierung von Forschungsdaten
Editoren	Panzer
Datum	15. April 2011
Dokument Status	Entwurf
Dokument Version	0.3.0

### Änderungen

Version	Datum	Name(n)	Kurzinfo
0.1.0	14.03.2011	Panzer	initialer Entwurf
0.2.0	04.04.2011	Panzer	alle wesentlichen Elemente enthalten
0.3.0	12.04.2011	Panzer	interne Begutachtung

<sup>1</sup>This work is created by the WissGrid project. The project is funded by the German Federal Ministry of Education and Research (BMBF).

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Fedora	5
1.2	iRODS	5
1.3	sonstige Komponenten	6
<b>2</b>	<b>Design</b>	<b>7</b>
2.1	Aufbau	7
2.2	Integrationsvariante: iRODS als Back-End für Fedora	7
2.3	Integrationsvariante: Fedora als Metadatenkatalog für iRODS	8
<b>3</b>	<b>Technische Umsetzung</b>	<b>12</b>
3.1	Art des Systems	12
3.2	Programmiersprachen	12
3.3	Datenspeicherung	13
3.4	Absicherungsmöglichkeiten der Systeme	13
3.5	Log files:	14
3.6	Performance und Skalierbarkeit	14
<b>4</b>	<b>Installation und Benutzung</b>	<b>15</b>
4.1	iRODS Installation	15
4.2	Fedora Installation	17
4.3	sonstigen Komponenten	17
4.4	Systeme starten	18
4.5	Erste Schritte	18
<b>5</b>	<b>Versionsinformationen</b>	<b>20</b>
5.1	Einleitung	20

5.2	Minimale Systemanforderungen . . . . .	20
5.3	Unterschiede zu Fedora-/iRODS-Originalauslieferungsversionen . . .	20
5.4	Implementiert . . . . .	20
5.5	Entwicklungsplan . . . . .	21
5.6	Bekannte Probleme/Schwächen . . . . .	21

## Hinweis:

Bei diesem Deliverable handelt es sich um eine prototypische Entwicklung, die noch nicht für den Produktivbetrieb geeignet ist.

Wegen eines internen Fehler im iRODS Rule-/Microservice-System wird das iCommand "imv" noch nicht unterstützt.

## 1 Einleitung

Eine Teilaufgabe des Unterprojektes „WissGrid AP3 - Langzeitarchivierung von Forschungsdaten“ ist die Integration der Systeme Fedora Commons und iRODS. Dieses Dokument beschreibt den Stand der Entwicklung und die Entwicklungsergebnisse werden dokumentiert.

Für die Integration wurden mehrere Module entwickelt, sowie Anpassungen an den Systemen iRODS und Fedora vorgenommen. Die wesentlichen Entwicklungen sind das iRODS-Akubra Modul auf Fedora Seite und die Erweiterung und Anpassung von Rules, sowie die Entwicklung von Microservices auf iRODS Seite. Zudem wurde mit dem Module "wissgridservices" ein SOAP-basierter Service entwickelt der eine weitergehende Verarbeitung vornimmt.

Ziel der Integration ist die Ausnutzung der jeweiligen Stärken der beiden Systeme, um hierüber eine umfassende Langzeitarchivierungs- und Nachnutzbarkeitsstrategie zu unterstützen. Die Verwaltung durch Fedora ermöglicht über seine Schnittstellen eine leichte und standardisierte Integration in andere Anwendungen - insbesondere auch virtuelle Forschungsinfrastrukturen.

Der konkrete Integrationstyp gemäß [Deliverable 3.5.2 - WissGrid-Spezifikation: Grid-Repository](#) ist Typ B. Bei diesem Integrationstyp wird das Grid und dessen Infrastruktur für die Speicherung genutzt.

Das exportieren der Fedora-Schnittstellen über verschiedene Technologien (REST/SOAP), die Integration externer Systeme quasi als "Event-Listener" über das Fedora Messaging System und die Navigierbarkeit von Objektbeziehungen machen Fedora zum idealen Daten- und Metadatenverwaltungssystem. Wegen der Ausrichtung auf das Abspeichern, die Rule-Engine, die bestehenden storage-orientierten Microservices und die diesbezügliche Erweiterbarkeit des Systems bietet sich iRODS als Storgelösung im Gesamtsystem an.

Mit dem iRODS-Akubra Modul wird es dem Akubra-Lowlevel-Storage Modul (Fedora Storage Ebene) ermöglicht, iRODS als Speicherungssystem zu nutzen. Fedora abstrahiert mit diesem Lowlevel-Storage Modul von der konkreten Storage-Lösung und überläßt Plugins die Anbindung an eine bestimmte Storage-Technologie. Das iRODS-Akubra Modul implementiert die geforderte Schnittstelle und ermöglicht Fedora damit die Kommunikation mit dem iRODS-System über das Lowlevel-Storage Modul.

Demgegenüber gibt es Anwendungsfälle bei denen eine direkte Interaktion mit dem iRODS-Server gewünscht ist, z.B. um ganze Verzeichnisse mit einem Aufruf zu übertragen oder die von iRODS angebotene Optimierung (parallelisierung) der Übertragung auszunutzen. Die Konsequenz aus dieser Nutzung ist, dass Aktivitäten die direkt auf dem iRODS-System stattfinden dem Datenverwaltungssystem Fedora mitgeteilt werden müssen. Diese Rückmeldung an das

Fedora System wird als "Callback" bezeichnet. Konkret läuft der Callback so ab, das bei Ereignissen im iRODS-Server Rules feuern, welche Microservices Aufrufen, die wiederum Aufrufe auf Fedora durchführen.

Bevor auf Einzelheiten eingegangen wird, soll zunächst in aller Kürze auf die eingesetzten Komponenten eingegangen werden.

## 1.1 Fedora

Die "Flexible Extensible Data Object Architecture" (Fedora) ist ein System zur Verwaltung dateibasierter oder per HTTP referenzierbarer Ressourcen bzw. Daten, welches es erlaubt Objekte zu klassifizieren, zu gruppieren, in Beziehung zu setzen, und mit Metadaten anzureichern. Es ist dabei formatneutral und ermöglicht die Schaffung von virtuellen Objekten, d.h. die Ableitung neuer Objekte auf Basis bestehender zur Laufzeit (über Services, sog. Dissaminators).

Das Fedora Digital Object (FDO) ist die zentrale, zu verwaltende Komponente im Fedora System und kann durch das FOXML Format beschrieben werden. Andere Beschreibungsformate sind: METS, ATOM oder ATOM Zip (jeweils in fedora-spezifischen Ausprägungen). Das FDO aggregiert Nutzdaten indem es Referenzen auf "Content-Objekte speichert (sog. Datastreams), es hält systembezogene, wie auch Dublin Core Metadaten vor, es kann über RDF-Einträge Beziehungen zu anderen Objekten definieren und selbst Ziel von Beziehungen sein, sowie eine einfache Veränderungshistorie pflegen. FDOs stehen mit einem oder mehreren Content-Modellen in Beziehung, worüber die Struktur eines FDO festgelegt werden kann und zugleich eine Klassifizierung ermöglicht wird. Über ein Content-Modell kann ein FDO mit Service-Definitions und Service-Deployments (Implementierung) angereichert werden, worüber die o.g. virtuellen Objekte realisiert werden. Für die Interaktion mit dem System steht eine RESTful- wie auch SOAP-Schnittstelle zur Verfügung.

Die Stärken des Fedora Systems sind seine Abstraktion von den zugrundeliegenden Objektformaten, d.h. seine Formatneutralität, die einfache Möglichkeit Objekte in Beziehung zu setzen (per RDF), die Integration einer Objekt Versionierung, sowie die einfache Anbindung externer Systeme z.B. zum Zweck der Indexierung und Suche oder zur Metadaten Sammlung (OAI-PMH) über den Fedora Messaging Service.

Siehe: <http://www.fedora-commons.org/>

## 1.2 iRODS

Das "Integrated Rule-Oriented Data System" (iRODS) ist ein Storage System, das eine verteilte, virtuelle Speicherung ermöglicht, wobei auf logischen Objekten und Strukturen operiert wird. Das Wissen über die verwalteten Objekte ist im iCAT (Katalog) einer relationalen Datenbank abgelegt. Das System kann über Rules und Microservices (C Routinen) angepasst oder erweitert werden - über Rules auch zur Laufzeit. Workflows können durch entsprechende Rule- und Microservice-Kombinationen abgebildet werden. Mehrere iRODS-Server bilden eine Zone und teilen sich einen iCAT. Zur Speicherung können verschiedene Storage-Ressourcen eingebunden werden, was mit Blick auf Lastverteilung, Verfügbarkeit und allgemein zur Replikation oder Bitstream-Preservation sinnvoll ist.

Die Besonderheit bzw. Stärke des iRODS-Systems ist seine Rule-Engine, welche das System anpassbar macht. Es ist bspw. möglich, beim Ingest eine Checksumme zu berechnen und ebenfalls persistent abzuspeichern. Über eine weitere regelmäßig aktivierte Rule kann dann eine neue (aktuelle) Checksumme des Objektes berechnet und mit der initialen verglichen werden. Bei Abweichung kann das fehlerhafte Objekt dann durch ein korrektes Replikat, das beim Ingest durch eine entsprechende Rule automatisch erzeugt und auf einer anderen Storage-Resource gespeichert wurde, ersetzt werden.

Siehe: <https://www.irods.org>

### 1.3 sonstige Komponenten

**wissgridservices** Es handelt sich hierbei um einen eigenständigen, SOAP-basierten Web-Service. Derzeit wird hierüber der Callback<sup>2</sup> und Metadatenextraktion realisiert.

**arcs Davis WebDAV Schnittstelle** Nicht von Fedora gemanagte Objekte<sup>3</sup> werden mit einer Referenz auf diese Objekte verwaltet. Da iRODS-Ressourcen über das Schema "irods://..." adressiert werden und Fedora nur File- und HTTP-basierte Referenzen unterstützt, bedarf es eines Mechanismus um diese Diskrepanz zu überbrücken. Die "arcs Davis WebDAV" Komponente realisiert für das iRODS-System die hierfür erforderliche HTTP-Schnittstelle und ermöglicht damit den Zugriff auf Objekte im iRODS-System per HTTP.

Siehe: <https://projects.arcs.org.au/trac/davis>

**GSearch und Fedora OAI Provider Service** GSearch (Fedora Generic Search Service) ist eine externe Komponente, die über das Fedora Messaging System eingebunden werden kann und mit welcher weitergehende Indexierung und Metadatenverwaltung durchgeführt werden kann.

Der Fedora OAI Provider Service ist eine Implementierung für das OAI-PMH Protokoll.

Beide Komponenten werden aktuell noch nicht eingesetzt, sollen aber in einer zukünftigen Version für die entsprechenden Aufgaben eingebunden werden.

Siehe:

<https://wiki.duraspace.org/display/FCSVCS/Fedora+Framework+Services>

---

<sup>2</sup>Der Callback meint die Rückmeldung von Aktionen auf iRODS-Objekten direkt über den iRODS-Server an das Fedora System

<sup>3</sup>Bei von Fedora gemanagten Objekten ist Fedora nicht nur für die Verwaltung der Objekte zuständig, sondern führt auch dessen Speicherung über seine Lowlevel-Storage-Ebene durch.

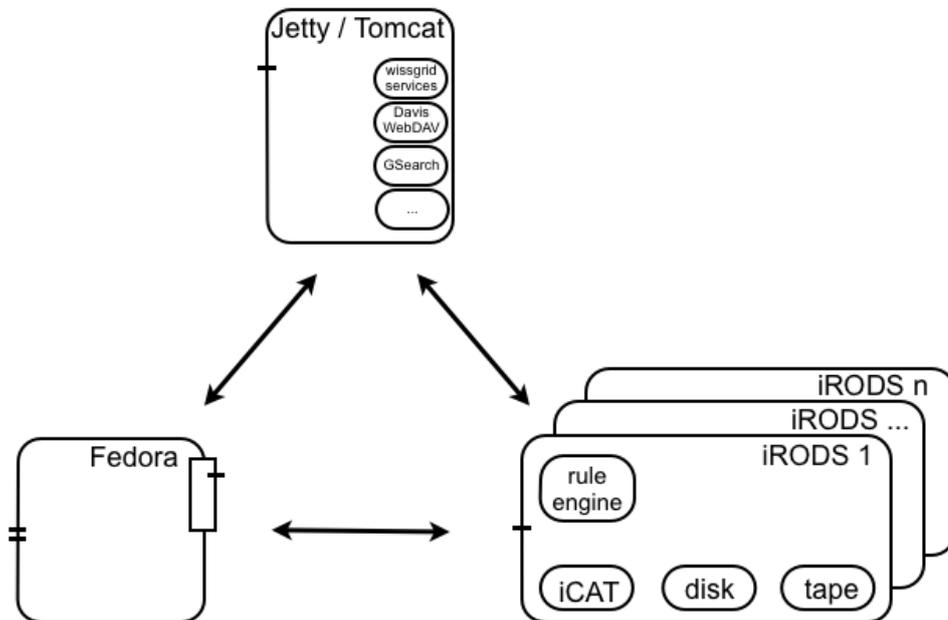


Abbildung 1: Systemkomponenten

## 2 Design

In diesem Unterkapitel wird der Aufbau und das Zusammenspiel des Systems besprochen.

### 2.1 Aufbau

Wie in der Einleitung bereits grob skizziert und in Abbildung 1 zu sehen besteht das entstehende System aus verschiedenen Komponenten, d.h. aus einem Fedora-Server<sup>4</sup>, sinnvollerweise mehreren iRODS-Servern, die zusammen eine Zone bilden und sich einen iCAT teilen. Zudem gibt es noch einige Zusatzkomponenten, wie das "wissgridservices" Modul, das "arcs Davis WebDAV" Modul und ggf. weitere optionale Module, so z.B. den Fedora Generic Search Service ( GSearch ).

Je nachdem wie das System genutzt wird, ergeben sich im wesentlichen zwei Integrationsvarianten, die nachfolgend beschrieben werden.

### 2.2 Integrationsvariante: iRODS als Back-End für Fedora

Bei dieser Integrationsvariante dient iRODS dem Fedora-System als erweitertes Storage System. Erweitert, weil das iRODS-System auch Aufgaben die über das reine Abspeichern hinausgehen, durch entsprechende Rules automatisch erfüllt. Als Beispiele seien die Replikation auf verschiedene Speicher-Geräte ( -Ressourcen ), die Checksummenberechnung und dessen regelmäßige überprüfung genannt.

<sup>4</sup>Bei einer Föderation oder einem Server-Cluster auch aus mehreren.

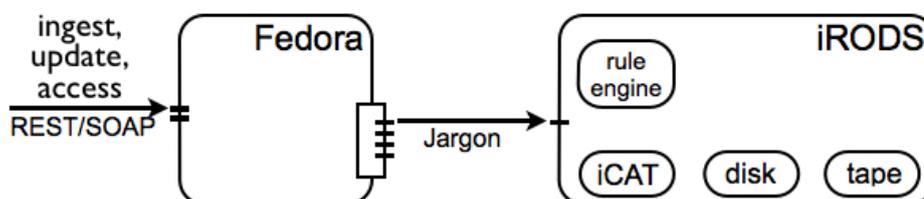


Abbildung 2: Integrationsvariante: iRODS als Back-End für Fedora

Die abzuspeichernden Datenobjekte ( Datastreams ) werden über das Fedora System, d.h. dessen Storage-Ebene abgespeichert. Im Fedora Jargon spricht man dabei von "managed Content", weil diese Objekte vollständig von Fedora verwaltet werden und nur über Fedora bezogen werden können<sup>5</sup>. Die Abspeicherung erfolgt über das Akubra-Lowlevel-Storage Modul, welches Fedora zur integrierten, d.h. über Fedora organisierten, Abspeicherung nutzt. Das Akubra-Lowlevel-Storage Modul abstrahiert dabei von der konkreten Technologie und nutzt einen Plug-in Mechanismus, über welchen Module für eine bestimmte Speicherungsart integriert werden können. Im vorliegenden Fall wird das iRODS-Akubra Modul ( Plug-in ) genutzt, um iRODS als Speicher-System verwenden zu können. Siehe Abbildung 2

Ein typischer Workflow sieht bei dieser Variante so aus, dass über die Fedora-API (SOAP- oder REST) ein Serviceaufruf vorgenommen wird und Fedora extrahiert dann ggf. Metadaten, stößt über sein Messaging-System externe Dienste an, und verarbeitet die Anforderung entsprechend.

Wenn es sich bei dem Serviceaufruf z.B. um eine Ingest-Anforderung handelt, so wird ein neues FOXML erzeugt, und in iRODS abgespeichert. FOXMLs sind immer "managed Content" d.h. Fedora ist für die Speicherung dieses Objektes verantwortlich. Es wird also das iRODS-Akubra Modul genutzt um eine Speicherung auf dem iRODS-System vorzunehmen. Wenn es sich bei dem Serviceaufruf jedoch um das Hinzufügen eines neuen Datastreams handelt, so besteht die Wahl zwischen den Kontrollgruppen ( Typen ) "managed", "redirect" und "external reference". Mit den Kontrollgruppen wird die Art bzw. Lokalität der Speicherung beschrieben, d.h. ob sich Fedora um die Speicherung kümmert oder ob das Objekt Fedora-extern gespeichert und lediglich referenziert wird.

### 2.3 Integrationsvariante: Fedora als Metadatenkatalog für iRODS

Hierbei wird ein Ingest vornehmlich über iRODS erfolgen und Fedora dient hauptsächlich als Datenverwaltungssystem. Der Zugriff ist sowohl über iRODS, als auch über Fedora möglich. Für diese Variante ist der bereits erwähnte Callback erforderlich, über welchen das Einfügen und Manipulieren von Datenobjekten direkt über den iRODS-Server an Fedora "zurückgemeldet" wird ( siehe Abbildung 3 ).

Bei der Abbildung zwischen Fedora Objekten und Verzeichnisstrukturen bzw. Dateien auf iRODS-Seite sind nachfolgend aufgeführte Organisationsformen zu unterscheiden. Mit dem

<sup>5</sup>Prinzipiell ist auch ein direkter Zugriff über iRODS möglich, jedoch ist diese wegen der von Fedora benutzten Speicherungs- und Verzeichnisstruktur nicht "komfortabel".

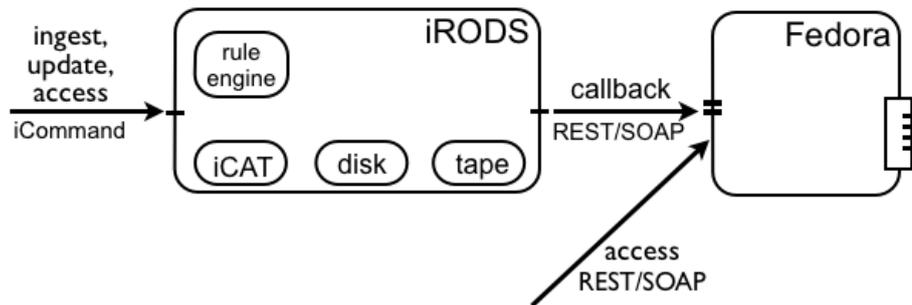


Abbildung 3: Integrationsvariante: Fedora als Metadatenkatalog für iRODS

Deliverable 3.5 wird die erste Variante unterstützt und die zweite ist in Bearbeitung.

- Jedes iRODS-Verzeichnis bildet ein FDO. Dateien innerhalb eines Verzeichnisses werden auf Datastreams dieses FDOs abgebildet. Enthaltene Verzeichnisse bilden ein neues FDO, welche mit dem übergeordneten durch eine RDF-Beziehung "fedora:isMemberOfCollection" verbunden sind. Dies setzt sich rekursiv fort.
- Jedes Datenobjekt ist zugleich ein FDO, das lediglich aus diesem einen Datenobjekt besteht.
- Der Umfang eines FDO variiert, z.B. kann es aus einer Gruppe von z.B. genau drei Datenobjekten und einer Beschreibungsdatei (METS, BAG-IT oder OAI-ORE) bestehen. Der Umfang zugehöriger Datenobjekte, deren Lokalität, Metadaten und Beziehungen werden über die Beschreibungsdatei festgelegt.

Die Adressierung von iRODS-Objekten erfolgt über das iRODS-Schema "irods://", welches von Fedora jedoch nicht unterstützt wird, da Fedora nur die Schema-Typen File und HTTP kennt. Es ist für Fedora Datenobjekte der Kontrollgruppen "Redirect" oder "External Reference" daher erforderlich, die in iRODS gespeicherten Objekte auch per HTTP referenzieren zu können. Die "arcs Davis WebDAV"-Schnittstelle (siehe Abbildung 4) realisiert eine HTTP-Schnittstelle auf das iRODS-System, über die ein HTTP-Zugriff auf iRODS Datenobjekte möglich wird. Die in Fedora referenzierten Datastreams bekommen als "Datastream location" die entsprechende HTTP-basierte URL eingetragen. Der Zugriff erfolgt beim "External Reference" mit einem Streaming durch Fedora, beim "Redirect" läuft das Streaming ausserhalb von Fedora ab.

Ein wesentliches Merkmal dieser Integrationsvariante ist das ein Ingest und Manipulationen an Datenobjekten nicht automatisch in Fedora registriert werden. Dies zu realisieren ist die Aufgabe des bereits erwähnten Callbacks. Er nimmt alle erforderlichen "weiterleitungen" an Fedora vor. Die Callback Implementation besteht aktuell aus:

- Nachfolgend aufgeführten Rules, die bei entsprechenden Aktionen feuern.
  1. acPostProcForPut - Nach Einfügen eines Datenobjekt
  2. acPostProcForCopy - Nach Kopieren eines Datenobjektes

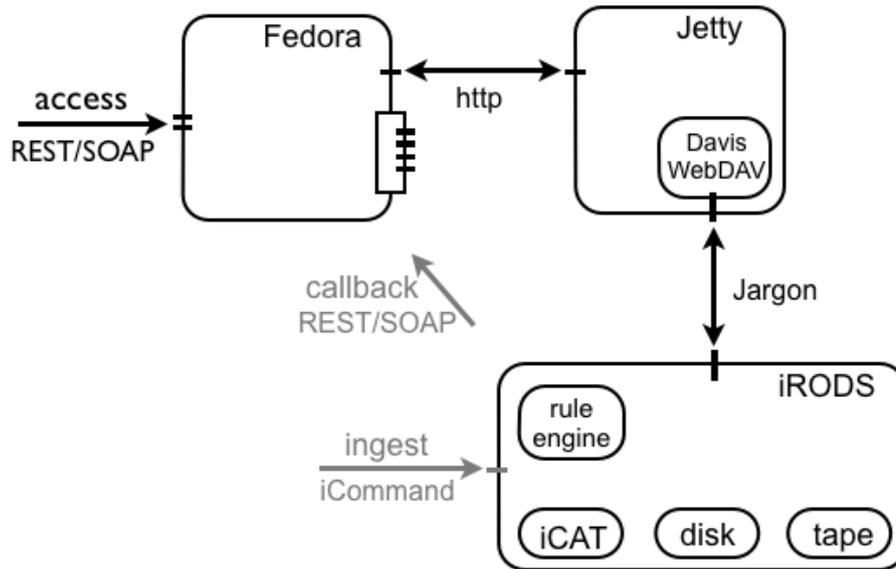


Abbildung 4: Einbettung der arcs Davis WebDAV Komponente

3. acDataDeletePolicy - Vor Löschung eines Datenobjektes
  4. acPostProcForCollCreate - Nach Erzeugung einer Collection
  5. acPreprocForRmColl - Vor Löschung einer Collection
  6. acPostProcForRmColl - Nach Löschung einer Collection
  7. acPostProcForObjRename - Nach Verschiebung/Umbenennung einer Collection bzw. eines Datenobjektes
- Die vorgenannten Rules delegieren dann die Arbeit auf nachfolgend aufgeführte Micro-services.
    1. msiWissGridCallbackForPutObject
    2. msiWissGridCallbackForCopyObject
    3. msiWissGridCallbackForDeleteObject
    4. msiWissGridCallbackForCollCreate
    5. msiWissGridCallbackForRmColl
    6. msiWissGridCallbackForObjRenameWithin
    7. msiWissGridCallbackForObjRenameFromOutside
    8. msiWissGridCallbackForObjRenameToOutside
    9. msiWissGridCallbackForRemoveCollAVUs
    10. msiWissGridCallbackForRemoveObjectAVUs
  - Die im "wissgridservices" Module enthaltenen Methoden implementieren die eigentliche Interaktion mit dem Fedora System.
    1. addDS

2. purgeDS
3. renameCollectionOrDSWithin
4. renameCollectionOrDSFromOutside
5. renameCollectionOrDSToOutside
6. ingestFDO
7. purgeFDO
8. addRS
9. purgeRS
10. removeCollAVUs

Ein typischer Workflow sieht so aus, dass ein iCommand an das iRODS-System gerichtet wird. Im Rahmen der Verarbeitung feuert eine Rule ( ggf. auch mehrere ) welche weitere Rules aktivieren und einen oder mehrere Microservices aufrufen kann. Ein Microservice stößt dann die weitere Verarbeitung an, d.h. nimmt einen Service-Aufruf auf Fedora-Seite vor oder geht den indirekten Weg über das "wissgridservices" Modul, welches die Weiterverarbeitung vornimmt.

Ein konkretes Beispiel wäre der iCommand "iput README.txt test/README.txt" welcher einen iRODS-Ingest mit der Datei "README.txt" im aktuellen Arbeitsverzeichnis vornimmt und die Datei unter dem Namen "README.txt" im iRODS-Verzeichnis "test" unter dem aktuellen Arbeitsverzeichnis auf iRODS-Seite speichert. Infolge dieses Speicherns wird die Rule "acPostProcForPut"<sup>6</sup> aktiviert. Diese Rule ruft den Microservice "msiSysChksumDataObj" auf, welcher eine Checksummenberechnung vornimmt und delegiert die Weiterverarbeitung an den Microservice "msiWissGridCallbackForPutObject", welcher den "Object-Path", den logischen<sup>7</sup> Speicherort für das Datenobjekt ermittelt. Der letztgenannte Microservice ruft dann die Methode "addDS" ( add-Datastream ) des "wissgridservices" Modul auf. Hier werden weitere Objekteigenschaften ermittelt, AVU Metadaten ( Attribut Value Unit ) mit dem iRODS-Objekt verbunden und letztendlich ein Service-Aufruf "addDS" im Fedora System vorgenommen.

---

<sup>6</sup>Es gibt auch eine entsprechende Pre-Rule, die vor dem Speichern aktiviert wird

<sup>7</sup>Durch die Unterscheidung von logischen und physischen Speicherorten ist es iRODS möglich, z.B. die Anzahl der max. zulässigen Dateien pro Verzeichnis zu beschränken, ohne dass dies in besonderem Maße beachtet werden muss.

## 3 Technische Umsetzung

### 3.1 Art des Systems

Beim entstehenden System handelt es sich um eine verteilte Client-Server Anwendung, die aus den nachfolgend aufgelisteten Komponenten besteht. Die Interaktion mit den Systemen erfolgt beim Fedora-System über dessen REST- oder SOAP-Schnittstellen (API-A für Access, API-M für Manipulation), beim iRODS-System erfolgt die Interaktion mit dem System über dessen Kommandozeilen Tools (iCommands) oder über die Java-basierte Jargon API. Das "arcs Davis WebDAV" Modul Realisiert eine Webschnittstelle, die mit dem Browser, WebDAV-Clients oder anderen Mechanismen zum Zugriff per HTTP genutzt werden kann. Fedora Kommuniziert mit dem iRODS-System über die Java-basierte Jargon API. Der Callback, d.h. die Rückmeldung von Ereignissen auf dem iRODS-System an das Fedora-System erfolgt über SOAP-basierte Webservices.

- iRODS-System

Das iRODS-System wird durch mindestens einen, meist aber mehrere iRODS-Server gebildet, welche eine Zone bilden. Die iRODS-Server einer Zone teilen sich einen iCAT (Katalog), dabei handelt es sich um eine relationale DB zur Speicherung von Metadaten, wodurch die Konsistenz des iRODS-Systems gewährleistet wird.

Im iRODS-System arbeitet man i.d.R. auf logischen Dateien und iRODS nimmt die Abbildung auf physische Dateien vor.

- Fedora-System

Das Fedora-System besteht zum einen aus einer Web-Anwendung, welche die Schnittstellen realisiert. Die Webanwendung delegiert Anforderungen dann an den eigentlichen Server.

- Weitere Services

- wissdrid-Services, d.h. Methoden zur Metadatenextraktion und zur Implementierung des Callback
- GSearch (zur Indexierung)
- OAI-PMH
- Fedora-Messaging

- arcs Davis WebDAV Schnittstelle

### 3.2 Programmiersprachen

- iRODS

- C für Microservices, Python für Microservices möglich
- Jargon API (Java) zur Interaktion mit dem iRODS-System
- iCommand, Kommandozeilen Tools zur Interaktion mit dem iRODS-System
- Fedora
  - Java
  - Restful und SOAP Web-Services zur Interaktion mit dem System:
    - API M - Management API zur Manipulation und
    - API A - Access API für den Zugriff
  - Jargon API (Java) zur Interaktion mit dem iRODS-System
  - Einfache Webschnittstelle zur Interaktion mit dem Fedora System
- arcs Davis WebDAV
  - java-basierte Webanwendung, implementiert WebDAV Schnittstelle für iRODS
- wissgridservices
  - java-basierter SOAP Web-Service

### 3.3 Datenspeicherung

- Es können mehrere, auch verschiedenartige Storage-Ressourcen eingebunden werden, wobei die wesentliche Voraussetzung hierfür, die "mountbarkeit" der Speicherungsgeräte im jeweiligen System ist.
- Fedora speichert die von Fedora selbst gespeicherten Objekte ("managed") über das Akubra-Lowlevel-Storage-Modul, d.h. in iRODS. Von Fedora referenzierte Objekte ("re-direct" oder "external reference"), d.h. externe Objekte zu denen lediglich eine URL zu deren Lokalität existiert, werden nicht direkt im Fedora-System gespeichert, sondern als externe Resource verwaltet. Um Operationen auf diesen Objekten auszuführen muss Fedora diese Objekte zunächst selbst laden.

### 3.4 Absicherungsmöglichkeiten der Systeme

- iRODS
  - Benutzer/Passwort
  - GSI
  - Kerberos in Entwicklung
  - ACL für Autorisierung
- Fedora
  - JAAS (Java Authentication and Authorization Service)
  - Benutzer/Passwort

- XACML aktueller Mechanismus für Autorisierung
- FeSL (Fedora Security Layer) für Authentifizierung und Autorisierung - für letzteres in Entwicklung
- arcs Davis WebDAV
  - Nutzt den iRODS-Mechanismus

Aktuell wird lediglich mit Admin- bzw. Robot-Usern gearbeitet, d.h. auf Basis von Benutzername und Passwort. Ziel der weiteren Entwicklung bzgl. der Sicherheitsaspekte wäre eine Erweiterung des Systems dahingehend, so dass eine Eins-zu-Eins Abbildung von Fedora- und iRODS-Benutzern realisiert wird. Ein weiteres Ziel wäre die Integration von GSI (Grid Security Infrastructure) - alternativ wäre auch Kerberos denkbar. Die weiteren Entwicklung wird auf Basis des JAAS-Framework (Java Authentication and Authorization Service) erfolgen.

### 3.5 Log files:

- iRODS
  - allgemeines Log: \$IRODS\_HOME/server/log/rodsLog.yyyy.mm.dd
  - empfangene Nachrichten: \$IRODS\_HOME/server/bin/RECV.log
  - gesendete Nachrichten: \$IRODS\_HOME/server/bin/SENT.log
- Fedora
  - allgemeines Log: \$FEDORA\_HOME/server/logs/fedora.log
- wissgridservices
  - allgemeines Log: \$CATALINA\_HOME/logs/catalina.out

### 3.6 Performance und Skalierbarkeit

Durch die direkte Interaktion mit iRODS System, d.h. durch Verwendung der Kommandozeilen Tools (iCommands) wird der Datentransfer automatisch optimiert, indem Datenpakete auf mehrere Threads und Socket-Verbindungen verteilt, parallel übertragen werden.

Eine Möglichkeit der Performanceverbesserung auf Fedora Seite wäre die Errichtung eines Server-Cluster, was jedoch aktuell nicht Teil der Entwicklung ist und somit hier nicht weiter diskutiert werden soll.

## 4 Installation und Benutzung

Die eigentliche Installation ist in der "Readme.txt" (im Installationsverzeichnis) beschrieben. Im folgenden werden darüberhinausgehende Informationen geboten.

Alle benötigten Ressourcen sind in dem ausführbaren Java-Archiv "WissGrid\_Deliverable3.5.jar" enthalten. Mit dem Aufruf "java -jar WissGrid\_Deliverable3.5.jar" wird der Installationsprozess gestartet (alternativ durch Doppelklick). Dabei werden die Dokumentation, die Installationsprogramme für iRODS und Fedora und eine Webservice-Komponente in das angegebene Zielverzeichnis kopiert. Fedora, iRODS und die Webservice Komponente sind entsprechend der "Readme.txt" separat zu installieren.

**Hinweis:** Es wurde sich gegen eine zusammenhängende "Alles-in-Einem" Installation entschieden, weil auf Installationsfehler teilweise nur durch eine Ausgabe wie "konnte Verzeichnis /lib nicht finden" reagiert wird. Solche Fehler, würden eine zusammenhängende Installation u.U. nicht abbrechen lassen, sie gefährden jedoch die Systemfunktion und sind später nur schwer zu finden. Daher wurde sich für die Schrittweise Installation entschieden. So hat man bei den Zwischenschritten die Möglichkeit die Ausgabe auf Fehlerhinweise zu kontrolliert, welche i.d.R. am Ende folgen.

### 4.1 iRODS Installation

Im Installationspfad liegt eine Datei "irods2.5-wissgrid.tgz", mit der entsprechend der Beschreibung in der "Readme.txt" verfahren wird.

Nach Abschluss der iRODS-Installation werden noch einige Hinweise gegeben, z.B.:

- **modify \$IRODS\_HOME/server/config/irodsHost**

In der letzten Zeile der Datei "\$IRODS\_HOME/server/config/irodsHost" ist das Kommentarzeichen zu entfernen und im Anschluss an "localhost" ist die IP-Adresse des Installationssystems anzugeben. Hiermit teilt man dem Server mit, welche IP-Adresse mit "localhost" assoziiert werden soll. Diese Information ist wichtig, wenn entfernt auf den Server zugegriffen wird. Ohne die Angabe kann es zu Fehlern kommen, bzw. der Client erhält keine Antwort.

Bsp.: "localhost 10.0.35.145"

- **update your PATH**

Z.B. in der ~/.profile folgendes hinzufügen:

```
export PATH= $IRODS_HOME/:$IRODS_HOME/clients/icommands/bin/:$PATH
```

- **irodsEnvFile anpassen, ~/.irods/.irodsEnv**

Die Datei muss mindestens folgendes enthalten, wobei die Werte natürlich angepasst werden müssen, so dass sie den Werten der Installation entsprechen:

- # iRODS server host name:  
irodsHost 'localhost'
- # iRODS server port number:  
irodsPort 1247
- # Default storage resource name:  
irodsDefResource 'demoResc'
- # Home directory in iRODS:  
irodsHome '/tempZone/home/rods'
- # Current directory in iRODS:  
irodsCwd '/tempZone/home/rods'
- # Account name:  
irodsUserName 'rods'
- # Zone:  
irodsZone 'tempZone'

Der iRODS-Server kann nun über ein Konsolen-Fenster mit "irodsctl start" gestartet werden. Durch Eingabe von "ils" wird dann ein Kommando (iCommand) an den iRODS-Server geschickt und dieses sollte eine Auflistung der Inhalte des im "~/.irods/.irodsEnv" angegebenen Arbeitsverzeichnis ("irodsCwd") erhalten. Damit dies aus jedem Verzeichnis heraus funktioniert, muss die Umgebungsvariable PATH die Pfadangaben zu "\$IRODS\_HOME/" und "\$IRODS\_HOME/clients/icommands/bin/" enthalten.

Wenn der iRODS-Server und Client für GSI konfiguriert wurde und ein Kommando mit einem Fehler "error while loading shared libraries: libglobus\_gss\_assist\_gcc64dbg.so.0: cannot open shared object file: No such file or directory" beantwortet wird, so muss die Umgebungsvariable PATH angepasst werden, z.B. indem der Datei "~/.profile" folgendes hinzugefügt wird:

```
export GLOBUS_LOCATION=/usr/local/globus
```

```
export LD_LIBRARY_PATH=
  $GLOBUS_LOCATION:
  $GLOBUS_LOCATION/lib:
  $LD_LIBRARY_PATH
```

Wenn man von einem Rechner aus, im Namen "verschiedener" Benutzern auf den iRODS-Server zugreifen will oder ggf. auf verschiedene iRODS-Server, so kann man entsprechende ".irodsEnv" Dateien<sup>8</sup> anlegen (z.B. .irodsEnv\_test1, .irodsEnv\_local, .irodsEnv\_remote). Ein Wechsel des iRODS-Accounts ist dann durch Setzen der Umgebungsvariable "irodsEnvFile" möglich, d.h.

<sup>8</sup>im Verzeichnis: "~/.irods/"

z.B. mit "export irodsEnvFile= ~/.irods/.irodsEnv\_remote". iCommands werden dann gemäß dieser speziellen Env-Datei (Konfiguration) durchgeführt und man erspart sich das ständige Ändern der Datei.

## 4.2 Fedora Installation

Bei der Datei "fcrepo-installer\_Wissgrid-AP3-Deliverable3.5.jar" handelt es sich um ein ausführbares Java-Archiv, das die Installation vornimmt. Die einzelnen Schritte sind in der "Readme.txt" aufgeführt,

### Fedora Installation

Durch nachfolgenden Aufruf wird der Fedora Installer ausgeführt (oder durch Doppelklick). Im Rahmen der Installation sind einige Angaben zu machen - für eine Testinstallation kann es im wesentlichen bei den Default-Werten bleiben. Wenn man mit den Konzepten besser vertraut ist, kann eine angepasste Installation oder nachträgliche Anpassung der Konfiguration vorgenommen werden. Zu Anfang wird man auf das Fedora-Messaging, GSearch oder SSL verzichten können.

### Konfiguration

- \$FEDORA\_HOME/server/config/fedora.fcfg  
Hierüber wird der Server konfiguriert.
- \$FEDORA\_HOME/server/config/akubra-llstore.xml  
Konfiguriert das Akubra-Lowlevel-Storage-Module.
- \$FEDORA\_HOME/server/config/fedora-users.xml  
Legt Benutzer und Rollen fest
- \$FEDORA\_HOME/server/config/jaas.conf  
Für die JAAS-Konfiguration
- \$FEDORA\_HOME/data/.../default/  
Hier liegen die Policy-Definitionen, mit welchen Rechte festgelegt bzw. geändert werden können.

### Log-Files

- \$FEDORA\_HOME/server/logs/fedora.log
- \$CATALINA\_HOME/logs/catalina.out

## 4.3 sonstigen Komponenten

### wissgridservices Modul Installation

Im wissgridservices Modul werden einige SOAP-basierte Web-Services realisiert. Die Installation z.B. in Tomcat erfolgt, indem die zugehörige JAR-Datei in das Verzeichnis "\$CATALINA\_HOME/webapps/" gelegt wird. Siehe "Readme.txt".

Durch Anpassung der Datei "\$CATALINA\_HOME/webapps/wissgridservices/WEB-INF/wissgridservices.properties" kann das Modul angepasst werden.

### **gSOAP Installation**

gSOAP wird im Rahmen der iRODS-Installation mit installiert, für weitere Hinweise sei auf die "Readme.txt" oder auf die Ausführungen in "[Web Services As Micro Services](#)" verwiesen.

**arcs Davis WebDAV Installation** Siehe hierzu:

- [Installation](#)
- [Konfiguration](#)

## **4.4 Systeme starten**

**Wichtig** Folgende Reihenfolge ist beim Starten des Systems einzuhalten!

**iRODS:** Der Server kann wie folgt gestartet, gestoppt oder neu gestartet werden:  
\$IRODS\_HOME/irodsctl {start | stop | restart}

**arcs Davis WebDAV:** Starten oder Stoppen des Servers erfolgt mit:  
/opt/davis/davis/bin/jetty.sh {start | stop | restart}

**Fedora:** Durch starten oder stoppen des Tomcat-Servers:  
\$CATALINA\_HOME/bin/{startup.sh | shutdown.sh}

## **4.5 Erste Schritte**

**Fedora:**

Es besteht die Möglichkeit das Fedora-System über seine Web-Schnittstellen, die Admin-Konsole oder von der Kommandozeile aus zu erkunden.

Web-Schnittstellen<sup>9</sup>:

- <http://localhost:8080/fedora>
- <http://localhost:8080/fedora/admin>

---

<sup>9</sup>Kann variieren, abhängig vom gesetzten Kontext-Pfad bei der Installation

Admin-Konsole:

- `$FEDORA_HOME/client/bin/fedora-admin.sh`

Aufrufe von der Kommandozeile (Rest-API)<sup>10</sup>:

- neues Objekt erzeugen mit:  
`curl -request POST "http://user:password@localhost:8080/fedora/objects/test:1?label=testwas"`
- Datastream laden:  
`curl -request GET "http://user:password@localhost:8080/fedora/objects/test:1/datastreams/MyDS/content" -o my.pdf`

## iRODS

Das iRODS-System kann über die iCommand Kommandozeilen-Tools getestet werden. Diese sind im Verzeichnis "`$IRODS_HOME/clients/icommands/bin/`" zu finden. Eine Hilfs-Ausgabe kann zu jedem iCommand mit der Option "`-h`" ausgegeben werden.

- Ingest: `iput einFile.txt a.txt`
- Collection anlegen: `imkdir a`
- Objekte im aktuellen Arbeitsverzeichnis auflisten: `ils`
- Hilfe: `iput -h`

## arcs Davis WebDAV

Das arcs Davis WebDAV-System kann z.B. über den Browser getestet werden<sup>11</sup>.

- `<host>:<port>/`
- `<host>:<port>/<zone>/home/<user>/`

---

<sup>10</sup>wieder abh. vom Kontext-Pfad

<sup>11</sup>Abh. von der Konfiguration, `<..>`-Angaben sind entsprechend zu ersetzen.

## 5 Versionsinformationen

### 5.1 Einleitung

Bei diesem Deliverable 3.5 handelt es sich um eine Prototypische Entwicklung, die noch nicht für den Produktivbetrieb geeignet ist. Sie zeigt den Stand der Entwicklung auf und es besteht ein konkreter Entwicklungs- und Verbesserungsplan.

### 5.2 Minimale Systemanforderungen

Da sich iRODS nur auf Linux-basierten Systemen installieren lässt, wird derzeit nur Linux unterstützt. Für Fedora (Tomcat) soll es später auch möglich sein, dieses auf einem anderen System installieren zu können, was aktuell wegen bestehender Systemabhängigkeiten noch nicht möglich ist.

Fedora benötigt ein JRE (Java Runtime Environment) in der Version 1.6.x.

### 5.3 Unterschiede zu Fedora-/iRODS-Originalauslieferungsversionen

- Im "irods2.5-wissgrid.tgz" sind neue Microservices und angepasste Rules enthalten, sowie einige Änderungen an der Konfiguration vorgenommen. Um den Installationsaufwand einfach zu halten, wurde sich für eine Auslieferungsvariante entschieden, mit welcher der iRODS-Server, incl. der Änderungen installiert wird. Zudem wird im Rahmen der Installation "gSOAP" installiert, um Microservices als Web-Service Clients realisieren zu können.
- Der "fcrepo-installer\_WissGrid-AP3-Deliverable3.5.jar" Installer enthält statt des Default Akubra-FS-Plugin (File-System), das iRODS-Akubra-Plugin. Zudem wurden einige Veränderungen an der Konfiguration vorgenommen.

### 5.4 Implementiert

Nachfolgend sind bislang implementierten Eigenschaften aufgeführt:

- iRODS-Akubra Modul, über welches sich Fedora mit dem iRODS-System verbindet. Fedora schreibt über dieses Modul Objekte der Kontrollgruppe "managed" in das iRODS-System bzw. leitet Manipulationen darüber weiter.
- Callback, d.h. das Einfügen und Manipulieren von Objekten über iRODS mit anschließendem Fortschreiben des Ereignisses auf Fedora. Hierbei können grundsätzlich drei Varianten unterschieden werden:
  1. Jedes iRODS-Verzeichnis (Collection) wird auf ein FDO abgebildet; jede enthaltene Datei ist auf ein Datastream dieses FDO abgebildet; Unterverzeichnisse bilden

ein neues FDO, das mit dem "übergeordneten" in einer "fedora:isMemberOfCollection" RDF-Beziehung steht.

2. Jede Datei wird auf ein FDO mit genau einem Datastream abgebildet.
3. Eine Gruppe von Dateien bildet eine FDO mit entsprechenden Datastreams. Die Zusammensetzung wird durch eine Beschreibungsdatei gemäß METS, OAI-ORE oder BAG-IT festgelegt.

Im Rahmen dieses Deliverables werden nur die erste Varianten unterstützt, die zweite befindet sich in Entwicklung und die letztgenannte ist Teil des weiteren Entwicklungsplanes.

## 5.5 Entwicklungplan

Für den weiteren Projektverlauf sind die folgenden Entwicklungen geplant.

- Die im vorausgegangenen Unterkapitel erwähnte Variante (3) mit zusammengesetzten Objekten gemäß METS, OAI-ORE oder BAG-IT.
- Verbesserung der Konfigurationsmöglichkeiten.
- Verbesserte Unterstützung für AuthN/AuthZ.
- Entwicklung eines GridFTP-Akubra Moduls, als weiteres Plug-in. Damit kann Fedora per GridFTP nutzbare Speicher-Ressourcen nutzen.

## 5.6 Bekannte Probleme/Schwächen

Im aktuellen Stand der Entwicklung sind die folgende Probleme bekannt und werden derzeit bearbeitet.

- Es ist lediglich eine Installation unter Linux möglich.
- Der iCommand "imv" aktiviert die iRODS-Rule "acPostProcForObjRename". Diese übergibt dem von ihr aufgerufenen Microservice ein Objekt "destObject", welches dem Zielobjekt entspricht. Im Rahmen einer Post-Rule sollte das Objekt vorliegen und mit Metadaten belegt werden können, doch dies ist nicht möglich und der Versuch führt zu Fehlern.
- Derzeit wird auf iRODS wie auf Fedora-Seite lediglich mit Robot- bzw. Admin-Usern gearbeitet.
- URLs und andere Benutzerdaten sind teilweise fest verdrahtet und sollen in Konfigurationsdateien ausgelagert werden.