



D3.5.3: Implementierung von Grid-Repositories

Teil 2: Federico

(Fedora Enabled Repository with Cocoon)

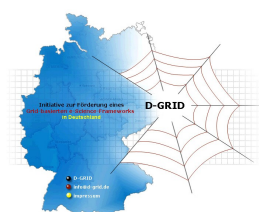
Version – 23.03.2012

Arbeitspaket 3

Verantwortlicher Partner - AWI

WissGrid

Grid für die Wissenschaft



Bundesministerium
für Bildung
und Forschung

WissGrid- Implementierung von Grid-Repositorien, Teil 2: Federico

Projekt: **WissGrid**

Teil des D-Grid Verbundes und der deutschen e-Science Initiative

BMBF Förderkennzeichen: 01|G09005A-G (Verbundprojekt)

Laufzeit: Mai 2009 - April 2012

Dokumentstatus: stabile Zwischenversion

Verfügbarkeit: öffentlich

Autoren: Bernadette Fritzsch, José Mejía (AWI)

Revisionsverlauf

Datum	Autor	Kommentare
11.03.2011	J. Mejía, B.Fritzsch	Erster Entwurf
14.04.2011	J. Mejía, B.Fritzsch	kleine Korrekturen. Punkt 9 (Wartung) wurde hinzugefügt.
20.04.2011	J. Mejía	Die Support E-mail Adresse und der Federico Homepage Link wurden aktualisiert.
23.03.2012	J. Mejía	Aktualisiert auf Federico 1.5

WissGrid- Implementierung von Grid-Repositorien, Teil 2: Federico

Inhaltsverzeichnis

1 Einleitung.....	4
2 Anforderungen.....	6
3 Anwendungsfälle.....	7
4 Systemanforderungen.....	12
5 Persistenz Inhalt-Modell.....	16
6 Installation.....	18
7 Testing.....	22
8 Aktueller Stand.....	24
9 Wartung und Support.....	25

1 Einleitung

Im Rahmen der in den vorhergehenden Deliverables¹ vorgestellten Spezifikation ist Fedora als eines der interessantesten Repositories diskutiert worden. Im vorliegenden Dokument wird gezeigt, wie Fedora genutzt werden kann für die Verwaltung von komplexen Metadaten. Insbesondere der Ingest der Daten mit der Erstellung der Metadaten wird durch die implementierte Lösung wesentlich vereinfacht.

Für Fedora Commons Repository 3.4 wurde unter dem Namen Federico (Fedora Enabled Repository with Cocoon) ein AJAX Frontend entworfen und implementiert für den Ingest von Metadaten in Anbindung an das C3Grid.

Für die Beschreibung von Daten in der Klimaforschung wird der ISO 19115 Standard verwendet, an dem sich die Archive orientieren. Allerdings ist das Profil recht umfangreich und komplex, so dass viele Nutzer bei der Erstellung der entsprechenden XML-Files überfordert sind. Hier setzen die im Rahmen von WissGrid entwickelten Tools an, indem sie die Metadatenerzeugung für den Wissenschaftler erleichtern. Als erstes Tool in der Workflow-Kette extrahiert JANEME aus den Fileheadern von NetCDF 3/4 und GRIB 1.0/2.0 Dateien die dort bereits abgelegten Informationen und sortiert sie in das ISO-konforme Profil ein.

Das ISO19115 Profil stellt exemplarisch eine komplexe Struktur in XML dar, dessen Bearbeitung von den Wissenschaftlern mit XML Editoren undenkbar ist. Das beschriebene Tool ist aber auch für andere Metadatenprofile geeignet, die Wahl von ISO19115 ist nur exemplarisch, um eine konkrete Implementierung bis zur Nutzbarkeit hin zu demonstrieren. Federico im Zusammenhang mit JANEME beschränkt diese schwierige Aufgabe auf wenige

¹ WissGrid-Spezifikation: Langzeitarchivierungsdienste, <http://www.wissgrid.de/publikationen/deliverables/wp3/WissGrid-D3.4.2-lza-dienste-spezifikation.pdf> sowie WissGrid-Spezifikation: Grid-Repository unter <http://www.wissgrid.de/publikationen/deliverables/wp3/WissGrid-D3.5.2-grid-repository-spezifikation.pdf>

WissGrid- Implementierung von Grid-Repositorien, Teil 2: Federico

Minuten durch die automatische Extraktion der Metadaten im gewünschten Format sowie die benutzerfreundliche Unterstützung während des Ingests in Fedora.

Dieses Deliverables bietet eine abstrakte Übersicht von Federico, geeignet als Startpunkt für Endbenutzer und Administratoren mit Hintergrund zu Fedora Commons, LZA oder Webapplikationen und soll durch eine übergreifende technische Dokumentation ergänzt werden, wo die Details zu der Implementation und Wartung unseres Systems ausführlich vertieft werden.

2 Anforderungen

- Das System soll so gestaltet sein, dass ein XSD Schema für die automatische Generierung eines Änderungsformulars in Federico übernommen werden kann. So wird das System generisch für den Einsatz anderer Metadatenprofile als ISO 19115.
- Der Benutzer soll Collectionen von Metadaten durch hierarchische Aggregation in Sets organisieren können, die das System durch eine entsprechende Semantik realisieren kann.
- Bereits aus anderen Quellen (z.B. der Metadatenextraktion per JANEME) vorhandene Metadaten sollen weitergenutzt werden. Dazu soll der Nutzer die Metadaten als eine XML Datei --*die konform zu dem oben genannten Schema ist*-- in einem Webformular hochladen können. Das System muss diese Informationen in die Felder eines Änderungsformulars automatisch übertragen (sogenanntes *Binding*).
- Die Metadaten müssen per Full-Text durchsuchbar sein und per OAI-PMH "geharvestet" werden können, um sie in bestehende Dateninformationssysteme integrieren zu können.
- Für die Authentifizierung sollten die bei den Nutzern in der Regel vorhandenen LDAP-Server der Heimateinrichtungen eingebunden werden können. Alternativ dazu soll auch die Authentifizierung über ein verwaltetes Open-ID-Konto möglich sein. Das System soll so konzipiert sein, dass andere Authentifizierungsmechanismen leicht hinzugefügt werden können.

3 Anwendungsfälle

Metadaten liefern eine Beschreibung der Daten und sichern die Nachnutzung von Datensätzen, indem sie Informationen über die Erstellung der Forschungsdaten und ihre Historie enthalten. Insofern können Metadaten sehr komplex sein und vom Umfang her vergleichbar oder sogar größer als die eigentlichen Daten. Für ihre Ablage soll Fedora genutzt werden, das im Rahmen von WissGrid als eine vielversprechende technische Umsetzung eines Repositories im Grid-Umfeld betrachtet wird. Die Nutzer des hier betrachteten Systems können unterschiedliche Rollen einnehmen (siehe Abbildung 1).

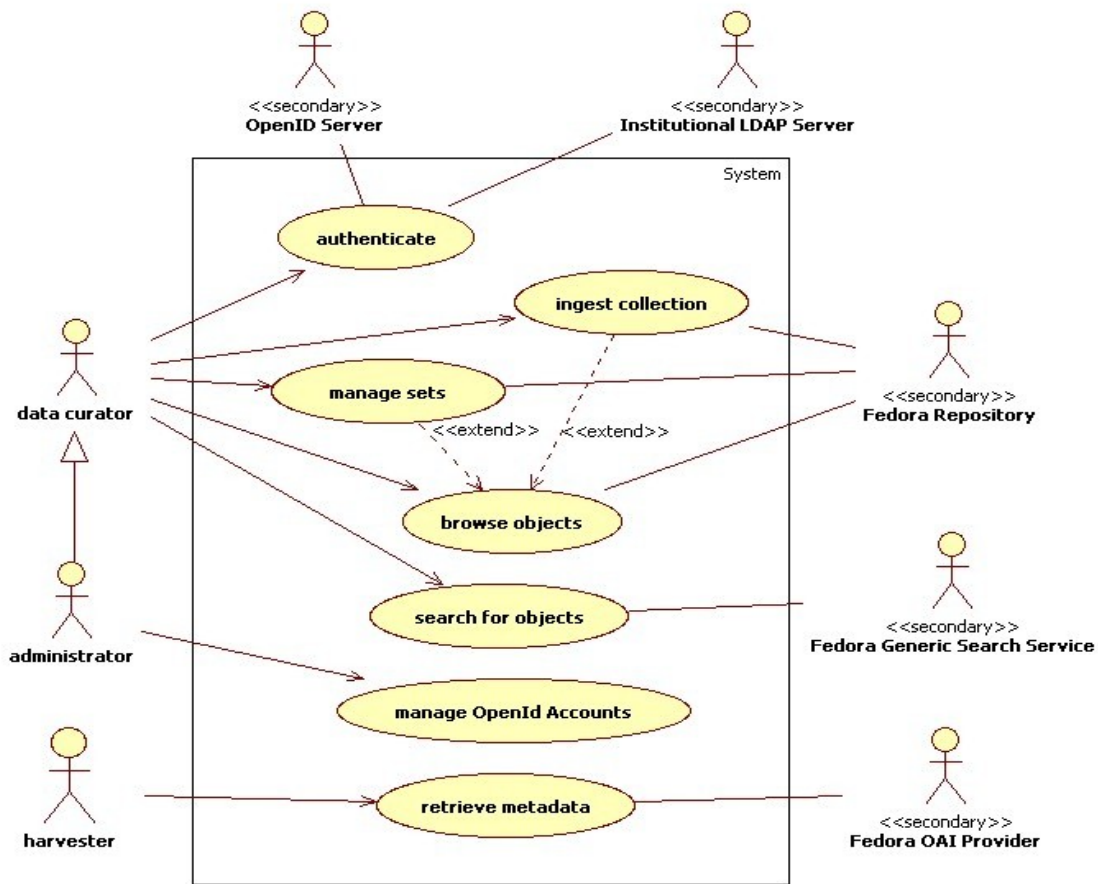


Abbildung 1: Übersicht über behandelte Use Cases

Hier soll vor allem der Datenkurator erläutert werden, da dies der mächtigste Use Case ist: Ein Wissenschaftler will neue Daten bereitstellen. Er muss sich authentifizieren, wozu es Schnittstellen zu den üblichen LDAP-Servern gibt. Zusätzlich wurde auch noch die Möglichkeit der Authentifizierung mittels OpenID integriert². Die Daten werden in einer vorgegebenen Hierarchie abgelegt, wozu der Nutzer die Datensätze in Sammlungen (Collections) anordnet.

Beim Einfügen neuer Collections lädt der Wissenschaftler ein vorher erstelltes XML-File mit der Beschreibung der Datensammlung hoch. Dieses wird dann von Federico validiert, wobei als Grundlage das C3Metadatenprofil³ dient. Das System ist aber so generisch angelegt, dass

² Diese Option ist insofern von Interesse, da in der Klimaforschung im Rahmen der Datenföderation zum 5. Sachstandsbericht des Intergovernmental Panel of Climate Change (IPCC AR5) OpenID genutzt wird und das vorliegende System dazu kompatibel sein sollte.

³ Das C3Grid Metadatenprofil ist ein ISO199115 basiertes Metadatenprofil mit einigen Anpassungen und Erweiterung, um im Rahmen des C3Grids unterschiedliche Datenquellen verbinden zu können. Siehe

neue Metadatenprofile integriert werden können. Falls das XML-File nicht valide ist, kann der Datenkurator es editieren und erneut einer Validierung unterziehen. Ist diese erfolgreich, so wird ein persistenter Identifier angefordert, ein Fedora Object FOXML generiert und diese dann in das Repository integriert, so dass das Objekt dann mit der Browse-Funktion gesucht werden kann. Über die interne OAI-Schnittstelle von Fedora können die Metadaten auch im Grid bereitgestellt werden, so dass sie dort verfügbar sind. Abbildung 2 zeigt die beim Ingest ablaufenden Prozesse schematisch.

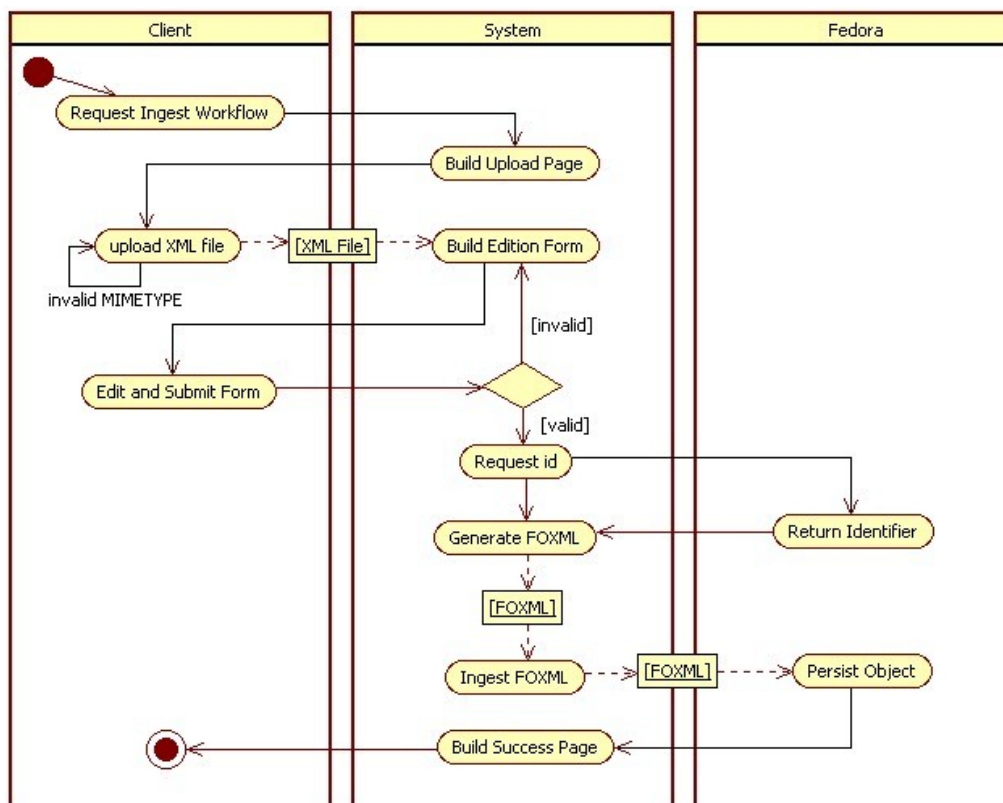


Abbildung 2: Ablauf beim Integrieren neuer Daten

Eine Zusammenfassung der betrachteten Anwendungsfälle liefert die folgende Tabelle 2.

<i>Anwendungsfall</i>	<i>Erklärung</i>
Authenticate	Dieser Anwendungsfall gewährt einem Datenkurator den Zugriff auf das System. Dieser Akteur gibt in das Login-Formular ein: (a) ein Paar Benutzername, Kennwort, das mit einem Benutzerkonto in einer institutionellen LDAP-Datenbank übereinstimmt, oder (b) einen Open-ID Benutzernamen, der in einer lokal verwalteten Datenbank registriert ist.
Browse Objects	Die Daten der Repository-Objekte, nämlich Sets und Collectionen, visualisiert der Kurator, als ob sie Entitäten eines Dateisystems wären.
Ingest Collections	Dieser Anwendungsfall ermöglicht einem Datenkurator die Aufnahme der Metadaten im Bezug auf eine Collection in das Fedora Repository.
Manage Sets	Dieser Anwendungsfall erlaubt einem Datenkurator die Erstellung, Änderung oder Löschung von Sets.
Search for Objects	Ein Datenkurator macht Full-Text Abfragen für die Dublin Core Eigenschaften der Objekte im Repository.
Manage Open-ID Accounts	Ein Supervisor kann eine Datenbank von Open-ID Benutzernamen verwalten. Diese Aufgabe umfasst das Einfügen, die Änderung und die Löschung der Benutzernamen und der dazugehörigen Attribute sowie die Zuordnung der Rollen.
Retrieve Metadata	Dieser Anwendungsfall erlaubt einem OAI-PMH Harvester, OAI Zugriffe auf das Repository zu senden.

Tabelle 1: Use Case Beschreibung

Die im hier dokumentierten System verwalteten Metadaten beschreiben Kollektionen von Datensätzen in externen Archiven mit Schnittstellen für verschiedene Protokolle wie HTTP, FTP und gridFTP und füttern einen in C3Grid bereits bestehenden globalen Lucene Index. So integriert sich Federico in eine Föderation aus verteilten Repositorien und stellt Information zur Verfügung, die mächtige Full-Textsuchen auf dem C3Grid Portal sowie komplexe Data-Mining Abschnittsberechnungen über die Kollektionen in Grid ermöglichen (siehe Abbildung 3).

WissGrid- Implementierung von Grid-Repositorien, Teil 2: Federico

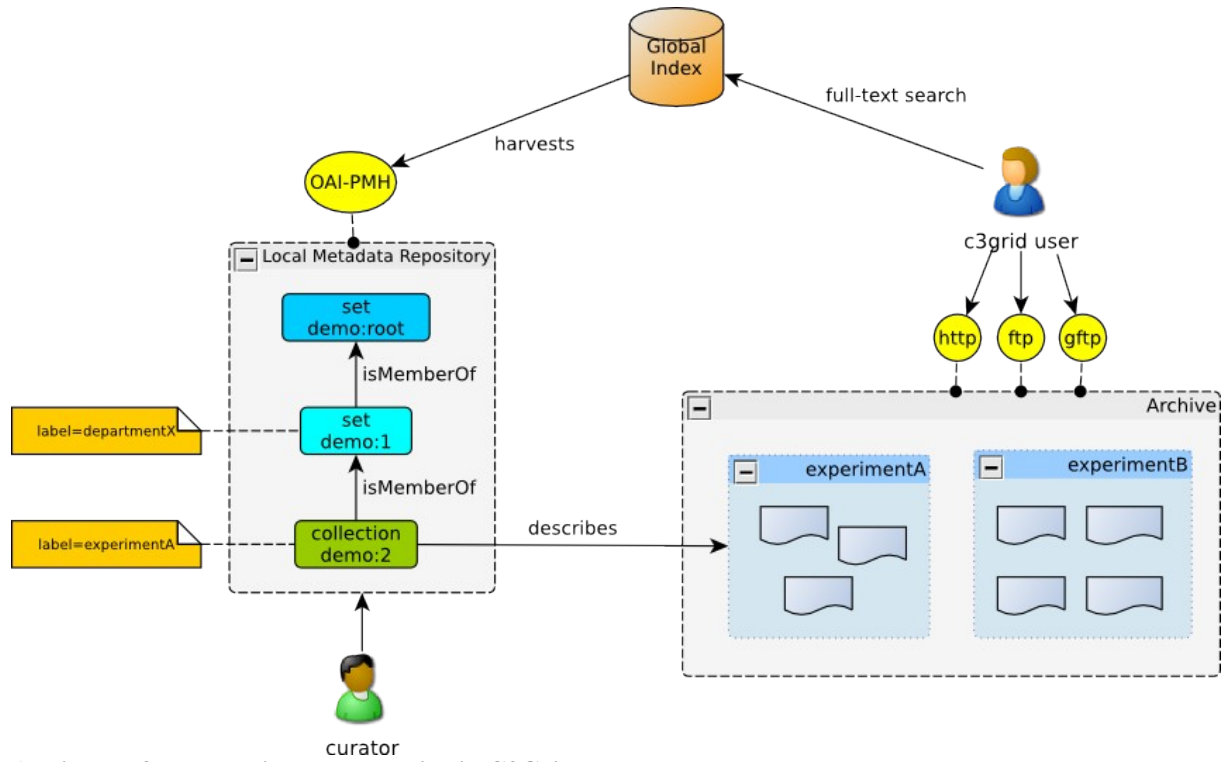


Abbildung 3: Integration von Federico in C3Grid

4 Systemanforderungen

Federico sollte auf einem Linux-Server mit mindestens 2 GB RAM Speicherkapazität und einer regelmäßig gesicherten großen Partition für die Hinterlegung der Daten und Indizes installiert werden.

Der Nutzer benötigt einen Javascript-enabled Browser, um mit der Webapplikation zu interagieren, wobei Mozilla Firefox 3+ und Google Chrome empfohlen werden. Die grafische Auflösung sollte in der Breite mindestens 1024 Pixel haben. Die Weboberfläche ist in 3 Sprachen verfügbar (Deutsch, Englisch und Spanisch), um den Einsatz in internationalen Kooperationen zu erleichtern.

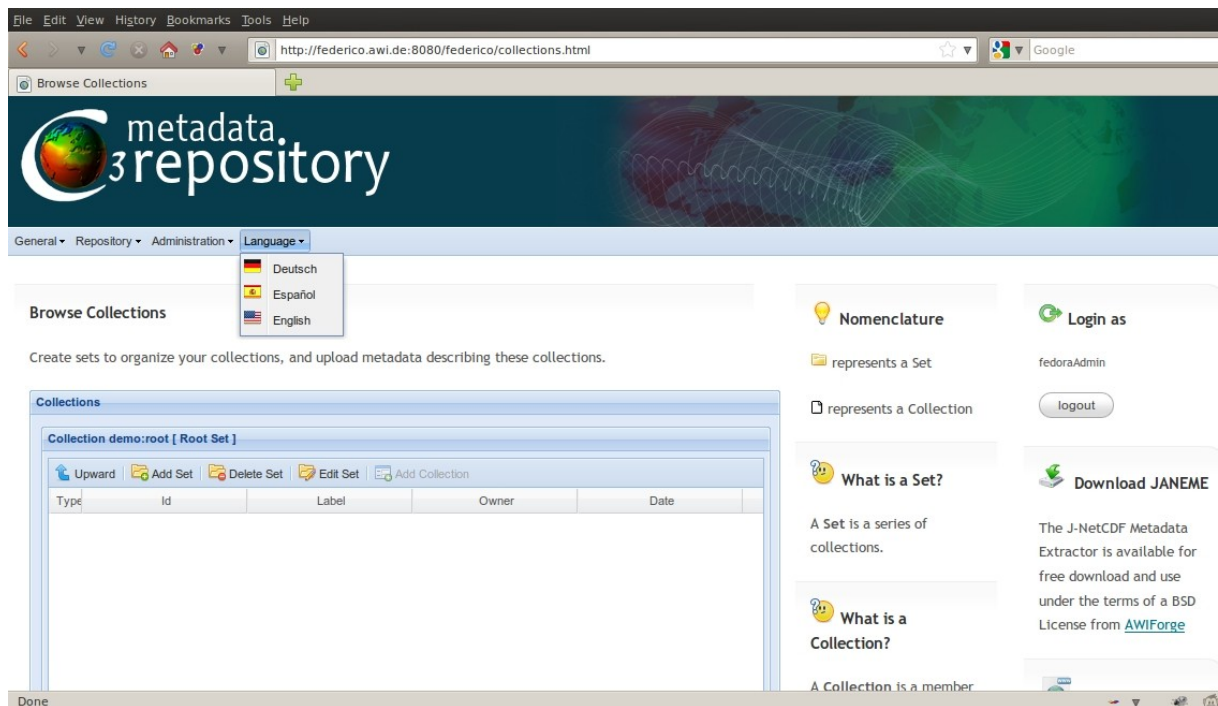


Abbildung 4: Screenshot - Sprachauswahl und Suche in Collections

WissGrid- Implementierung von Grid-Repositories, Teil 2: Federico

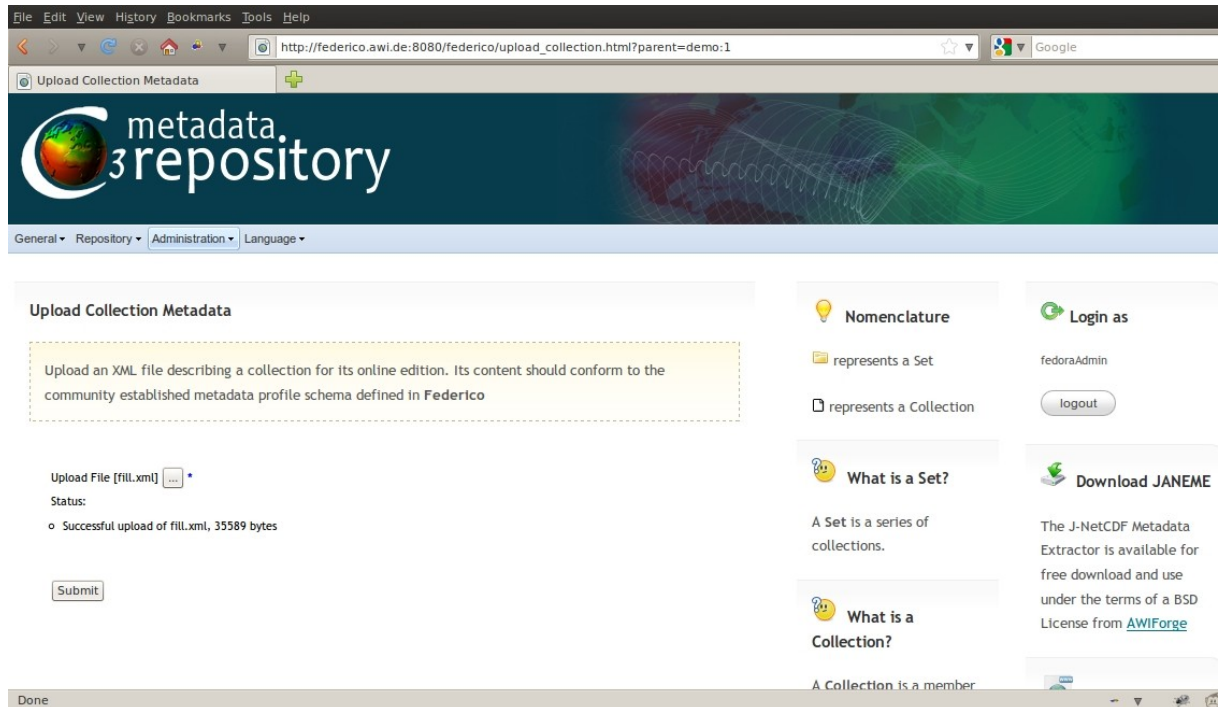


Abbildung 5: Screenshot - Hochladung einer XML Datei

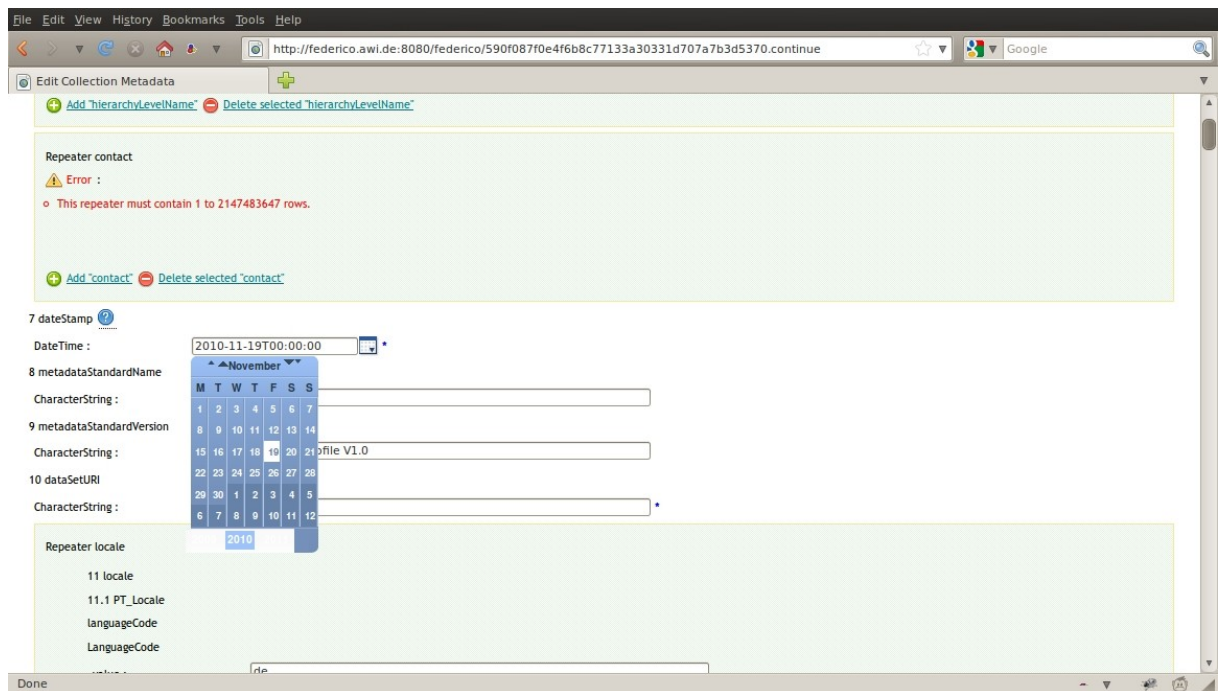


Abbildung 6: Screenshot - Metadaten Änderungsform

WissGrid- Implementierung von Grid-Repositories, Teil 2: Federico

Die Applikation läuft auf einem Tomcat Servlet Container 6.0 und wird durch folgende Frameworks angetrieben:

- Apache Cocoon 2.2
- Apache Solr 1.3
- Spring Framework 2.5
- Fedora Commons Service Frameworks

Apache Cocoon ist dabei der „core“ Framework der Webanwendung, auf dessen Grundlage die Entwicklung der Seiten und Workflows durch XML Technologie ermöglicht wurde. Die Applikationen sind in Modulen (in Cocoon Terminologie sogenannte „Blocks“) strukturiert, die sich auf beliebige URL-Pfade mounten lassen. Für den End-Anwender ist die gesamte technische Komplexität vollkommen transparent.

<i>Kategorie</i>	<i>Software</i>
Datenbank	3 MySQL Datenbanken: fedora3, proai, openid
Webanwendungen	Fedora Commons 3.4.1, Fedora Generic Search, Fedora OAI Provider, Solr 1.3
Web Framework	Apache Cocoon 2.2, Spring Security 2
Programmiersprachen	Oracle/Sun JDK 1.6, Groovy 1.7
Servlet Container	Apache Tomcat 6.0
Javascript Bibliotheken	ExtJS 3.2, DOJO
Build Manager	Maven 2.2
Server Betriebssystem	Linux
Web Browser (Klient)	Jeder Browser mit aktiviertem JavaScript. Mozilla Firefox 3.6+ ist bevorzugt.

Tabelle 2: Wichtige Software-Anforderungen

WissGrid- Implementierung von Grid-Repositories, Teil 2: Federico

Das folgende Diagramm stellt die Architektur Federicos und seine Beziehung zu anderen Webanwendungen, die auf dem gleichen Servlet-Container eingesetzt werden, dar.

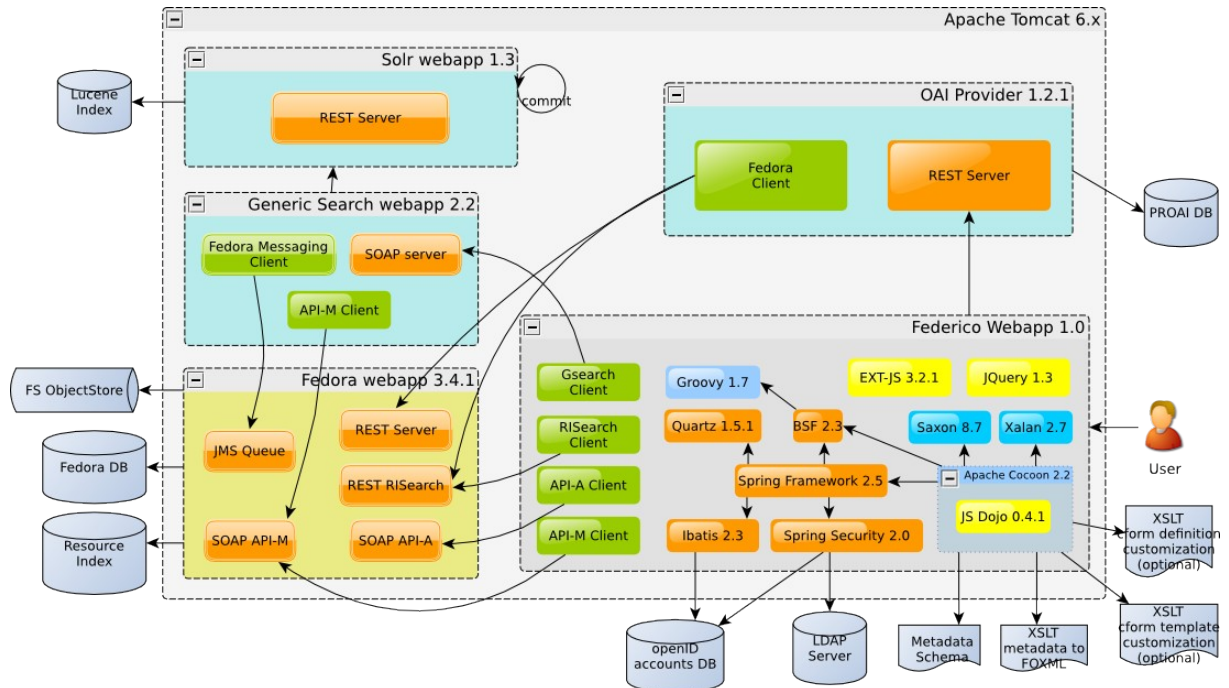


Abbildung 7: Architektur von Federico

5 Persistenz Inhalt-Modell

Fedora 3 identifiziert vier Hauptklassen für Objekte (Content-Modell, Datenobjekt, Service-Definition und Service-Deployment), auf deren Basis die Definition neuer Klassen möglich ist. Federico nutzt diese Möglichkeit und erweitert Fedora um zwei Content-Modelle, welche Sets und Collections definieren. Beziehungen in der Modell-Hierarchie werden durch RDF-basierte Einträge in den jeweiligen REL-EXT Datastreams der zugehörigen Datenobjekte abgebildet. Dies gilt ebenfalls für die Objektebene, d.h. Collections drücken über RDF-Einträge aus, zu welchem Set sie gehören (isMemberOf). Sets können wiederum die Zugehörigkeit zu einem übergeordneten Set ausdrücken.

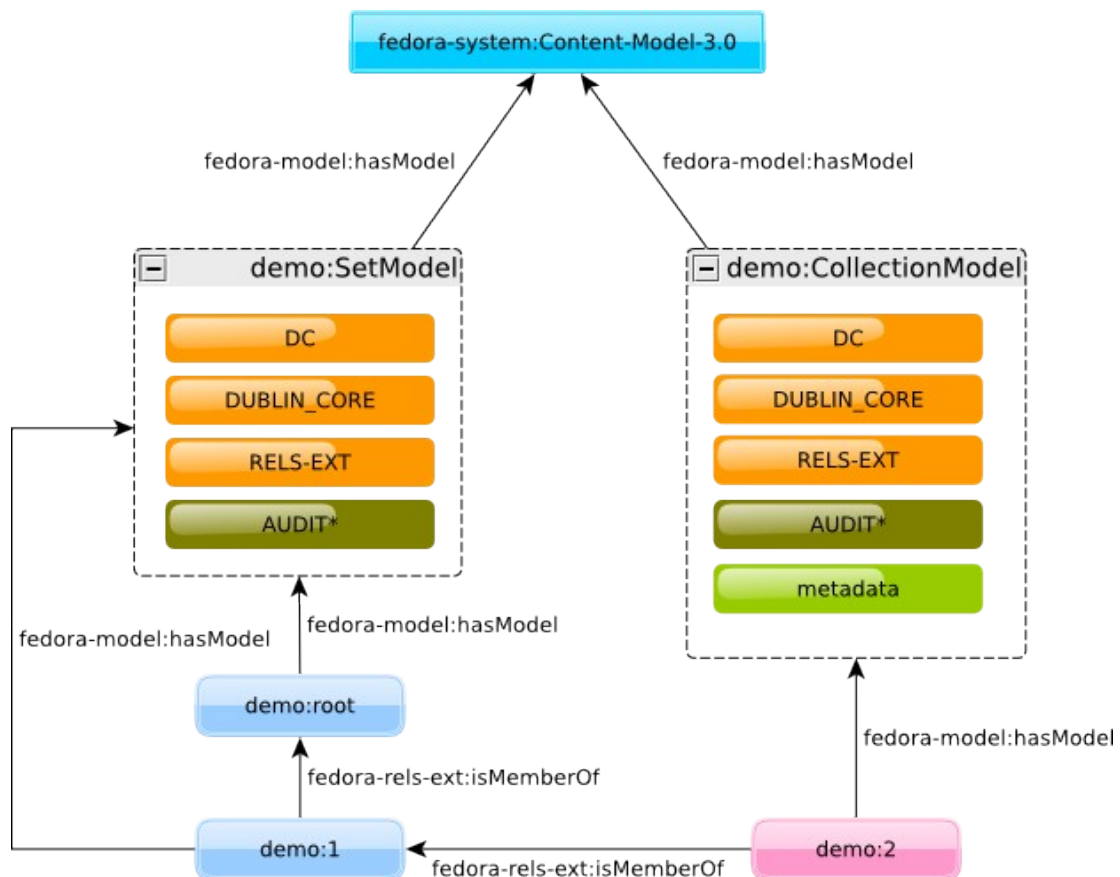


Abbildung 8: Content Modell von Federico

Die Abbildung 8 zeigt beispielhaft eine Collection `demo:2`, die Member des Sets `demo:1` ist. `demo:2` ist konform zu dem Modell `demo:CollectionModel` und beinhaltet die Datastreams

DC (Dublin Core) und DUBLIN_CORE⁴, RELS-EXT, AUDIT und *metadata*. In letzterem werden die gesammelten Metadaten gespeichert. Das Set *demo:1* ist konform zum Modell *demo:SetModel*, dieses kennt jedoch nur die default Datastreams; der Datastream *metadata* entfällt hier, da Sets Collections aggregieren sollen und selbst keine Daten verwalten. Zu unserer Abstraktion eines Filesystems gehört auch die Existenz eines obersten Sets, das in der Abbildung als *demo:root* bezeichnet wird.

Das Konzept eines Content-Modells in Fedora ermöglicht auch das Hinzufügen neuer Services zu einem Modell. Diese neuen Services sind ohne weiteres Hinzutun für alle diesem Modell entsprechenden Objekte aufrufbar. Es bedarf dazu keiner Änderung an den bestehenden Objekten. Im aktuellsten Stand von Federico finden solche Services jedoch keinen Einsatz.

⁴ Das Datastream *DUBLIN_CORE* wurde in Federico 1.5 eingeführt, um die Ergänzung der 15 klassischen DC Kernfeldern durch neue XML Elemente zu ermöglichen.

6 Installation

Für eine benutzerfreundliche und reibungslose Installation des Gesamtsystems hat das AWI ein Installationspaket basierend auf IzPack 4.3⁵ entwickelt. IzPack ist eine One-Stop-Lösung⁶ mit einer Apache-Lizenz 2.0 für die Verpackung, Verteilung und Bereitstellung von Anwendungen für die Java-Plattform.

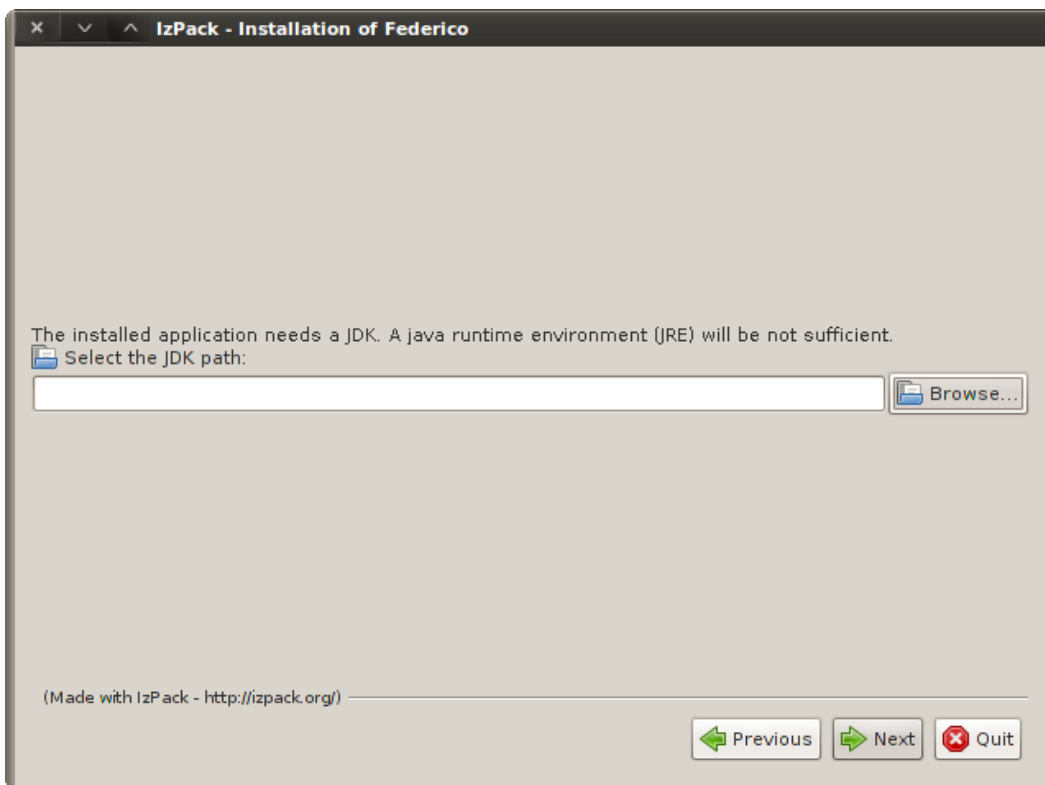


Abbildung 9: Prüfung des JDK Pfades

Dieses Installationspaket erfordert eine Java virtuelle Maschine und kann sowohl in einer Konsole als auch mit einer grafischen Oberfläche bedient werden. Während der Installation werden die JDK Version und wichtige Konfigurationsparameter gefragt oder überprüft wie z.B. den Pfad des JDK 1.6 im Dateisystem, Kennwörter, den Namen des Solr Core-Index, das gewünschte Fedora ID Namespace und die MySQL Datenbank- und LDAP-Anbindungsparameter, den Installationsordner usw.

⁵ <http://izpack.org/>

⁶ Unter dem englischen Begriff *One-Stop* bezeichnet man die Bereitstellung einer Reihe von Dienstleistungen oder Waren an einem einzigen Ort.

Federicos Installationspaket wurde programmatisch erweitert, um wenn gewünscht, die Neuerstellung der Datenbanktabellen durchzuführen.

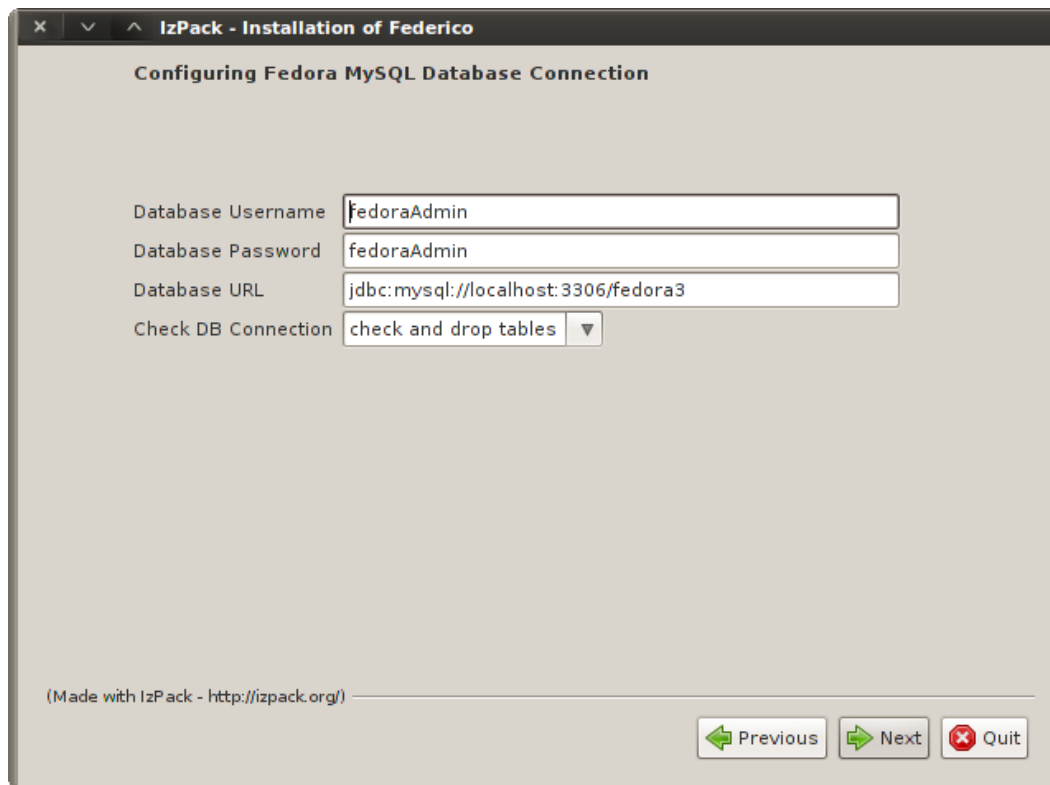


Abbildung 10: JDBC Datenbank Konfiguration

Optional ist die Installierung von extra Applikationen wie OpenDS 2.2.0 (einem Java-basierten LDAP Server), JMeter 2.4 (einer Java-Desktop-Anwendung zur Prüfung funktionaler Tests und zur Messung der Lastleistung einer Applikation), Luke 1.0.1 (einem handlichen Entwicklungs- und Diagnose-Tool, das einen bereits bestehenden Lucene-Indizes für die Ansicht seines Inhalts zugreift) und Visual VM 1.3.1 (einem visuellen Tool Integration verschiedener JDK Kommandozeilen-Tools und leichter Profiling-Funktionen) möglich.

WissGrid- Implementierung von Grid-Repositorien, Teil 2: Federico

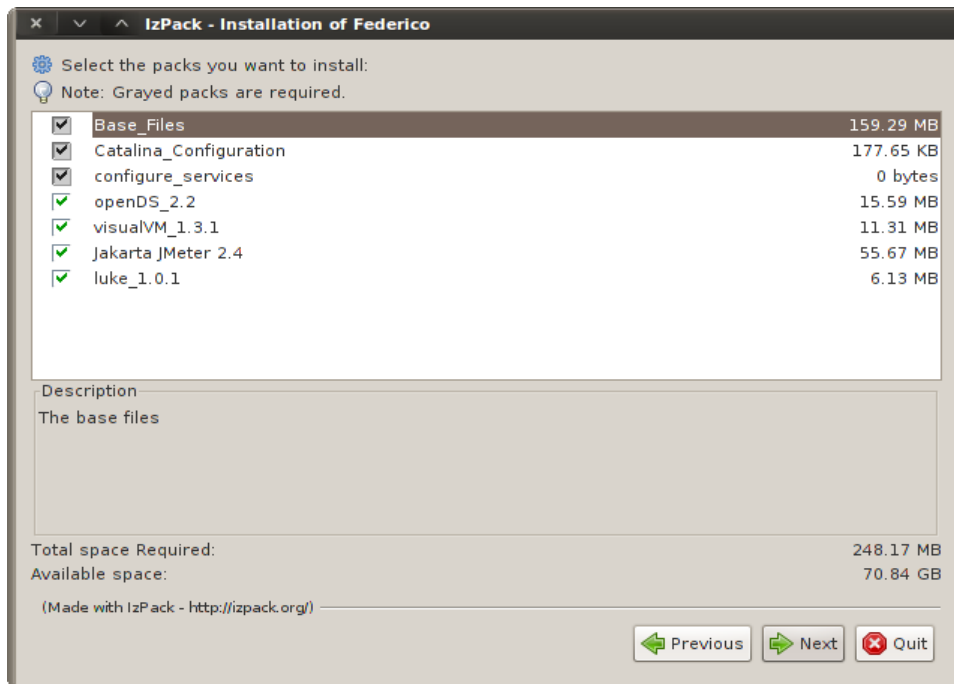


Abbildung 11: Auswahl der Installationspakete

Als Ergebnis der Installation wird ein voll funktionsfähiger Apache Tomcat 6.0 Container mit den nötigen Konfigurationen und Webanwendungen (Fedora, PROAI, Solr, GSearch und Federico) angelegt. Zu beachten ist, dass Federico wichtige Objekte wie Content-Modelle nach 10 Minuten seines erstens Laufs kreiert.

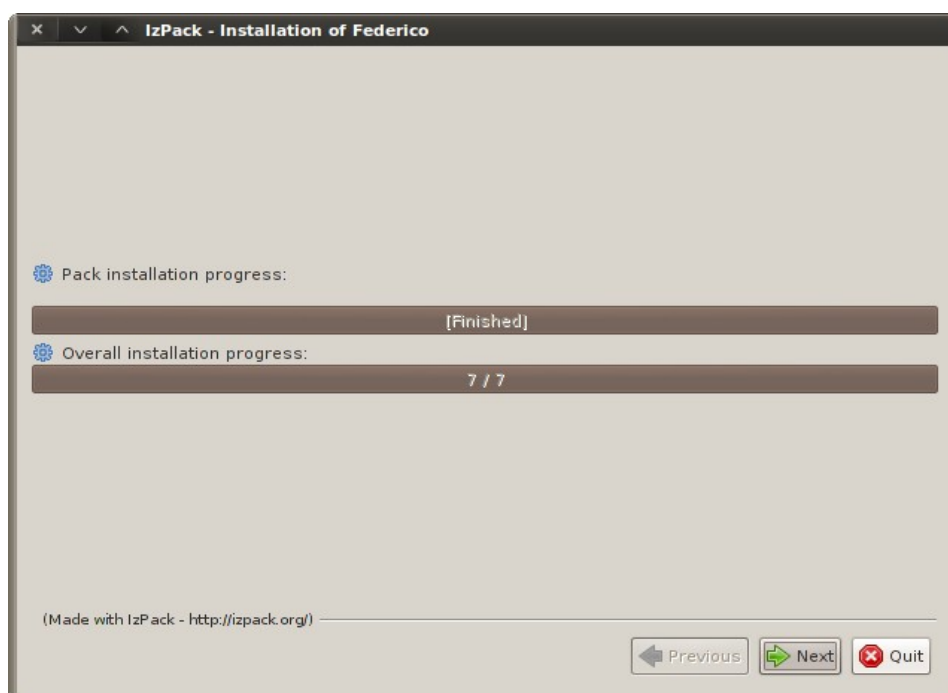


Abbildung 12: Installationsprogressbar

WissGrid- Implementierung von Grid-Repositorien, Teil 2: Federico

Das Skript `$federico_home/apache-tomcat-6.0.29/bin/init-script.sh` dient als Startup und Shutdown Programm für das Gesamtsystem. So kann Federico mit dem Aufruf

```
$federico_home/apache-tomcat-6.0.29/bin/init-script.sh[start | stop]
```

gestartet bzw. gestoppt werden.

Außerdem wird Tomcat Container als *default* konfiguriert, um HTTP-Requests unter dem URL <http://localhost:8080/federico> zu bedienen.

Falls notwendig, kann der Administrator das konfigurierte XSD Schema nach Bedarf anpassen oder durch ein eigenes XSD Schema ersetzen. Wie dieses Vorgehen und weitere Konfigurationen erfolgen und wie der Endnutzer mit dem System umgehen kann, wird in der offiziellen technischen Dokumentation von Federico beschrieben, die auf der Projecthomepage <http://aforge.awi.de/gf/project/federico/> verfügbar ist.

7 Testing

Das Installierungspaket legt die Datei *test-suite.jmx* im Root Verzeichnis der Distribution Federicos an. Sie besteht aus einer XML-Datei, deren Parameter während der Installation gesetzt werden und die als Input für JMeter 2.4 dient.

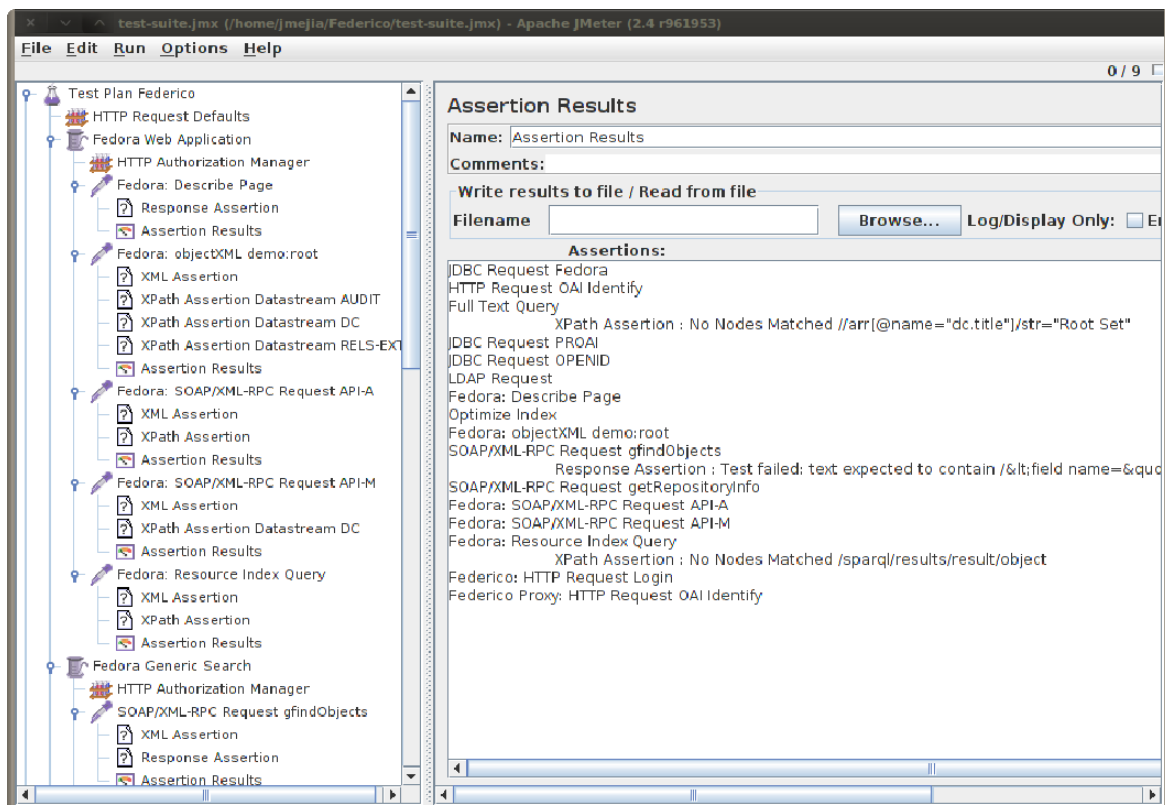


Abbildung 13: Screenshot der Assertionergebnisse in JMeter

Diese Testsuite überprüft folgendes:

- die Fedora Services API
 - API-A und API-M SOAP Anfragen
 - Resource Index REST Anfragen
- Fedora Framework Services
 - GSearch SOAP
 - PROAI Rest Identify Anfrage

WissGrid- Implementierung von Grid-Repositories, Teil 2: Federico

- Solr
 - Full-Text Suchanfrage
 - Reaktion auf Optimierungsbefehle über REST
- JDBC Datenbankverbindungen
- LDAP-Anfrage
- Erreichbarkeit der Federico Webanwendung
 - Login als Administrator
 - als PROAI Proxy

8 Aktueller Stand

Das produktionsfähige Release Federico 1.5 wurde am 27.01.2012 veröffentlicht. Seine technische Dokumentation sowie sein Sourcecode und Installer befinden sich auf dem AWI Projektmanagementserver unter dem URL <http://aforge.awi.de/gf/project/federico/>.

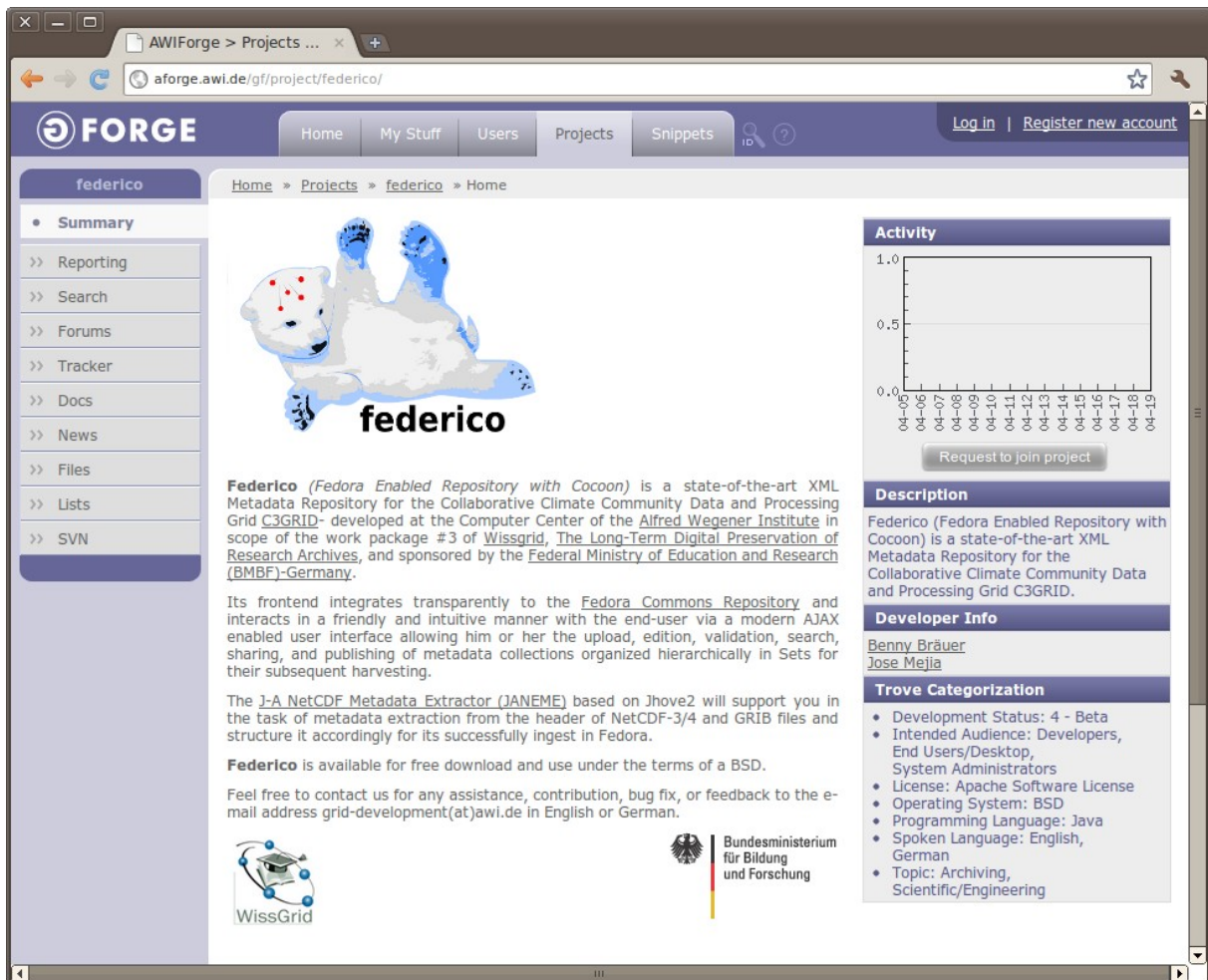


Abbildung 14: Screenshot der Federico Homepage

9 Wartung und Support

Das Modulcode wird von José Mejía Villar, M.Sc., vom Alfred-Wegener-Institut für Polar- und Meeresforschung in Bremerhaven, Deutschland gepflegt.

Der Entwickler kann für Supportanfragen unter <mailto:grid-development@awi.de> erreicht werden. Hier erhalten Sie fachliche Unterstützung in allen Fragen rund um Federico und zur Fehlerbehebung und können uns Feedback auf Englisch oder Deutsch geben.