# METAFOR OAI system deployed
# METAFOR Deliverable 4.2 M24

| **PROJECT** | |
|---|---|
| Project acronym | METAFOR |
| Project full title | Common Metadata for Climate Modelling Digital Repositories |
| Grant agreement no: | 211753 |
| Funding Scheme | Combination of Collaborative Projects & Coordination and Support Actions |
| Call Topic | INFRA-2007-1.2.1 Scientific Digital Repositories |
| | |
| **DOCUMENT** | |
| Deliverable | D4.2 Month 24 |
| Deliverable Title | OAI system deployed |
| Document Identifier | METAFOR-D4.2 |
| Date | April 2010 |
| Work Package | WP4 |
| Authors | BADC |
| Document Status | Final |
| Document Link | http://metaforclimate.eu/documents |

| **Dissemination Level** | | |
|---|---|---|
| PU | Public | |
| PP | Restricted to other programmes participants | |
| RE | Restricted to a group specified by the Consortium | X |
| CO | Confidential | |

| **Document History** | | | |
|---|---|---|---|
| Version | Date | Comment | Author/Partner |
| 0.1 | April 13th 2010 | First draft from contributions from C.Pascoe, M.Morgan and H. Ramthun | S.A.Callaghan/BADC |
| 0.2 | April 20th 2010 | Major Corrections | BN Lawrence/BADC |
| 0.3 | July 13th 2010 | Final Version | BN Lawrence/BADC |

## Abstract

Metafor originally aimed to deploy a document moving system for metafor based on the open archive information system protocol for metadata harvesting (OAI/PMH). This has been deprecated in favour of exploiting document feeds based on the Atom protocol (RFC3287). Atom feeds have been deployed at two partner sites within metafor and are being used for international data transfer.

# 1 Introduction

This document is the Metafor deliverable D4.2 – OAI system deployed. Figure 1 shows the original basic architecture diagram as proposed in the original Metafor description of work.

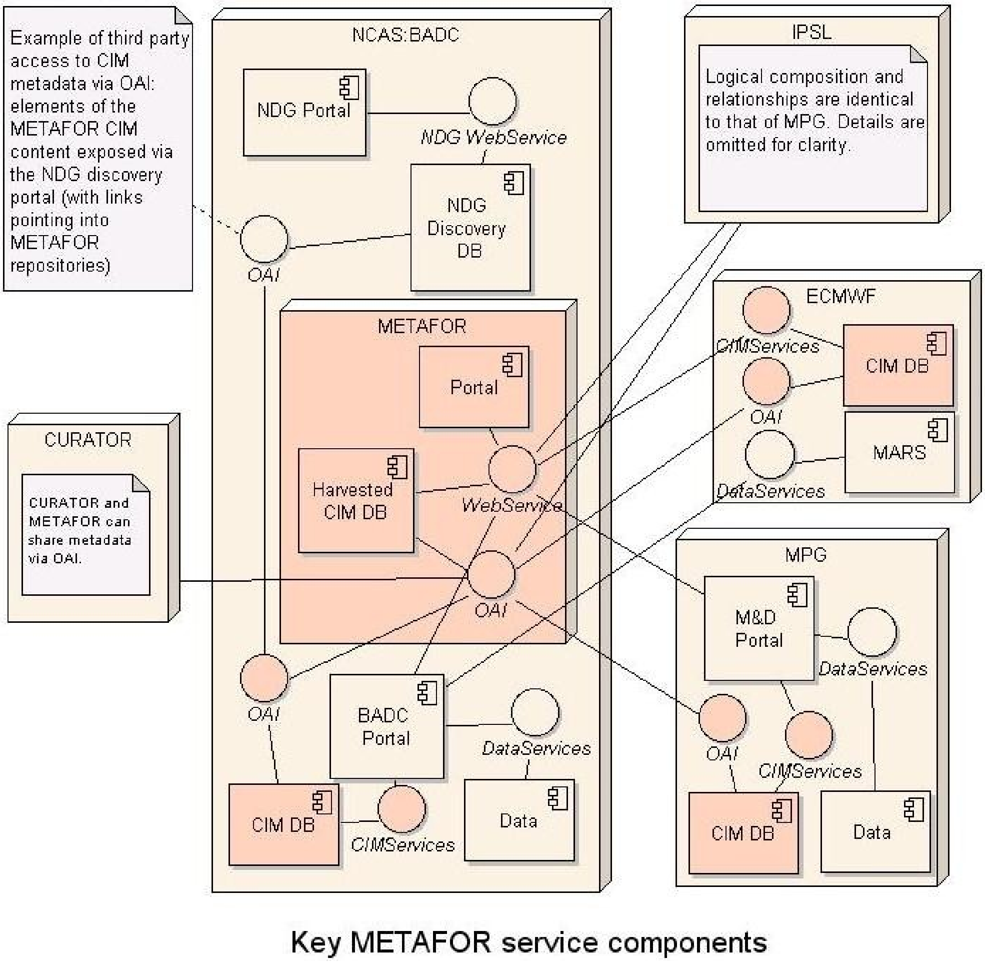

Example of third party access to CIM metadata via OAI: elements of the METAFOR CIM content exposed via the NDG discovery portal (with links pointing into METAFOR repositories)
NCAS:BADC
NDG Portal
NDG WebService
NDG Discovery DB
OAI
METAFOR
Portal
Harvested CIM DB
WebService
OAI
OAI
BADC Portal
DataServices
CIM DB
CIMServices
Data
IPSL
Logical composition and relationships are identical to that of MPG. Details are omitted for clarity.
ECMWF
CIMServices
CIM DB
OAI
MARS
DataServices
MPG
M&D Portal
DataServices
OAI
CIMServices
CIM DB
Data
CURATOR
CURATOR and METAFOR can share metadata via OAI.
Key METAFOR service components

Figure 1: Key Metafor service components.

(Note that data services are out of scope for Metafor per se, but are in scope for IS-ENES, and a key activity in WP4 and the services group in Metafor is the interaction with IS-ENES, and the interaction with the Earth System Grid Curator team as part of CMIP5.)

In this introduction, we discuss the deliverable in three sections: a brief description of the requirements and our progress with OAI, what we decided to dowhat we expect, the relationship/dependencies between Metafor work packages, and the expected dependencies between metafor and IS-ENES where they exist[1].

**Initial Services**
Early in the project we had settled on the use of Geonetwork as a tool for managing CIM documents, and because Geonetwork (now) comes with an OAI/PMH interface, we expected to use this as the main tool for data moving.

[1] Please note that there are NO expected dependencies on IS-ENES from Metafor.

Both BADC and MPIM deployed OAI interfaces to repositories during test phases, using both Geonetwork and other tools. In all cases except Geonetwork, the method of exploiting OAI/PMH exporting documents into a bespoke OAI/PMH product. In this mode, some difficulties were encountered with OAI/PMH between the different implementations.

Even as Metafor was working with OAI/PMH, the Atom syndication format has become established commercially, and in academia, and in peer EC projects (for example, GENESI-DR). Some of the Metafor team experimented with Atom in other contexts, and found it preferable to OAI/PMH. It became clear that as well as being easier to use within Metafor, the use of Atom would improve exposure of Metafor information into other projects. Accordingly, Metafor experimented with Atom feeds as as mechanism for moving data, and then moved to using Atom preferably. So this deliverable describes how Metafor is deploying Atom infrastructure, rather than OAI/PMH.

This deliverable has been delayed since not only has Metafor moved from OAI/PMH to Atom, but the central role of the questionnaire in the project meant that a true equivalent of the original deliverable would be the use of Atom in moving data within the project and between the project and the ESG team. This has now been accomplished.

In the remainder of this deliverable, we introduce atom feeds, the atom structure, and the way they are being used within Metafor.

## 2 Atom Feeds

Metafor has decided not to deploy OAI/PMH as the main method for moving content (as originally planned). It was decided that the Atom publishing framework is simpler & more flexible than the OAI framework, particularly in the light of the consideration that the Metafor system architecture is predicated on RESTful services.

**What are RESTful web services?** A RESTful web service is a simple web service implemented using HTTP and the principles of REST. It is a collection of resources with three defined aspects:

1. The base URI of the seb-service (e.g.www.metaforclimate.eu/cim-repository);
2. The MIME type of the data supported by the web-service (e.g. XML, JSON, TEXT);
3. The HTTP operations supported by the web-service (e.g. POST, PUT, DELETE, GET).

Although we could have met our OAI prototype deployment milestones, and deploy an OAI based system, we will instead deliver all the OAI functionality for this project using the Atom standard. Advantages of this decision are as follows:

1. OAI/Geonetwork cannot cover all the data integration scenarios whereas Atom can;
2. There is extensive Atom support in all major programming languages & across all major operating systems;
3. Atom promotes a style of architecture (RESTful) which has been demonstrated to be particularly adapted to the web;
4. Atom promotes a loosely coupled event driven architecture which has been demonstrated to be an important aspect of successful enterprise application integration scenarios;

5. Atom inherently allows HTTP caching which is an important performance optimisation technique in distributed systems;
6. Atom supports discovery scenarios in which one item within a feed can link to another feed.
7. Atom can support multiple content-types (i.e. XML, JSON, TEXT).

## 3 Atom structure

Atom is fully documented in RFC3287, this section provides a brief summary of relevance to Metafor:

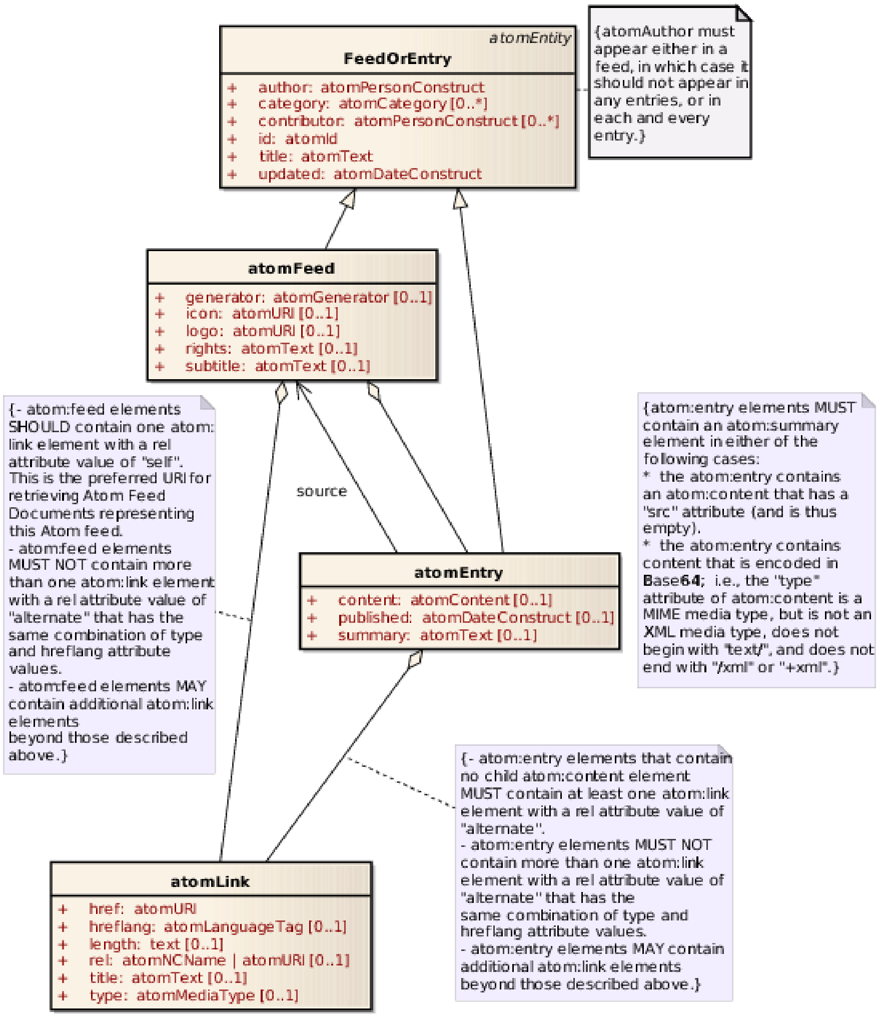
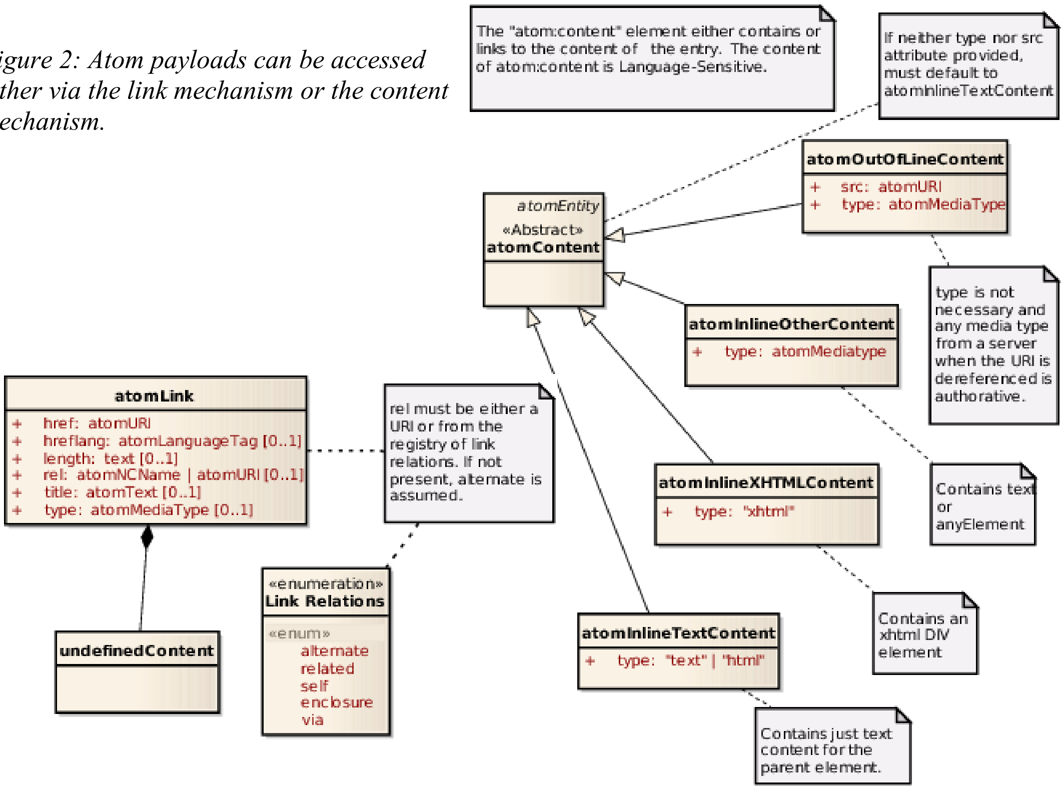

atomEntity
FeedOrEntry
+ author: atomPersonConstruct
+ category: atomCategory [0..*]
+ contributor: atomPersonConstruct [0..*]
+ id: atomId
+ title: atomText
+ updated: atomDateConstruct
{atomAuthor must appear either in a feed, in which case it should not appear in any entries, or in each and every entry.}
atomFeed
+ generator: atomGenerator [0..1]
+ icon: atomURI [0..1]
+ logo: atomURI [0..1]
+ rights: atomText [0..1]
+ subtitle: atomText [0..1]
{- atom:feed elements SHOULD contain one atom:link element with a rel attribute value of "self". This is the preferred URI for retrieving Atom Feed Documents representing this Atom feed.
- atom:feed elements MUST NOT contain more than one atom:link element with a rel attribute value of "alternate" that has the same combination of type and hreflang attribute values.
- atom:feed elements MAY contain additional atom:link elements beyond those described above.}
source
atomEntry
+ content: atomContent [0..1]
+ published: atomDateConstruct [0..1]
+ summary: atomText [0..1]
{atom:entry elements MUST contain an atom:summary element in either of the following cases:
* the atom:entry contains an atom:content that has a "src" attribute (and is thus empty).
* the atom:entry contains content that is encoded in Base64; i.e., the "type" attribute of atom:content is a MIME media type, but is not an XML media type, does not begin with "text/", and does not end with "/xml" or "+xml".}
{- atom:entry elements that contain no child atom:content element MUST contain at least one atom:link element with a rel attribute value of "alternate".
- atom:entry elements MUST NOT contain more than one atom:link element with a rel attribute value of "alternate" that has the same combination of type and hreflang attribute values.
- atom:entry elements MAY contain additional atom:link elements beyond those described above.}
atomLink
+ href: atomURI
+ hreflang: atomLanguageTag [0..1]
+ length: text [0..1]
+ rel: atomNCName | atomURI [0..1]
+ title: atomText [0..1]
+ type: atomMediaType [0..1]

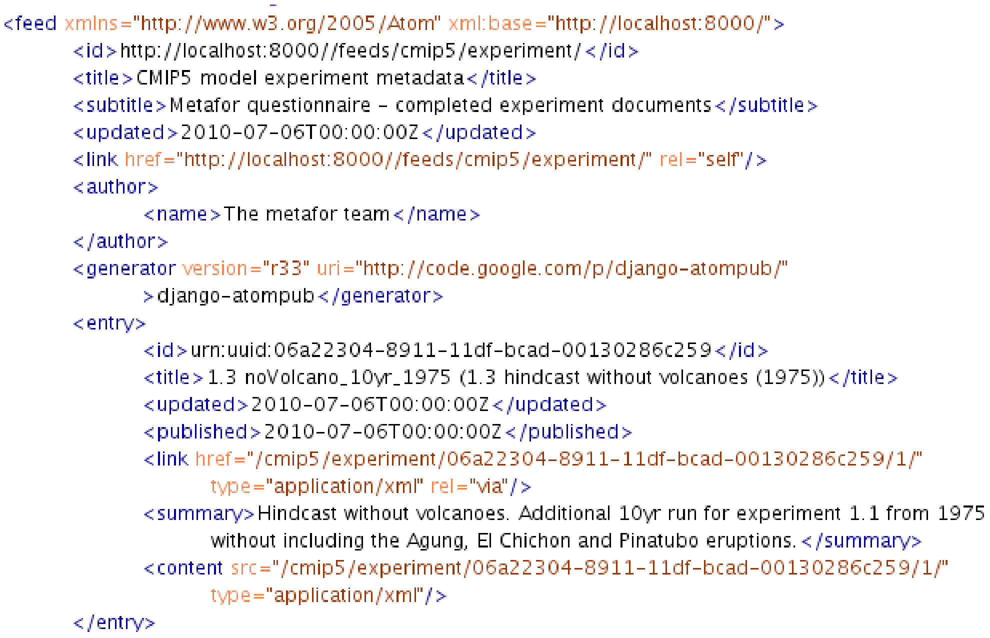
*Figure 1: UML showing the key characteristics of atom feeds and entries*

The key characteristics from a Metafor point of view is that Atom feeds consist of lists of entries which can carry XML payloads either internally or externally (of which more below). The feeds can distinguish naturally between new, updated, and unchanged entries, so that a client can simply provide a date and get all new and updated entries since that date.

*Figure 2: Atom payloads can be accessed either via the link mechanism or the content mechanism.*


The "atom:content" element either contains or links to the content of the entry. The content of atom:content is Language-Sensitive.
If neither type nor src attribute provided, must default to atomInlineTextContent
atomEntity
«Abstract»
atomContent
atomOutOfLineContent
+ src: atomURI
+ type: atomMediaType
type is not necessary and any media type from a server when the URI is dereferenced is authorative.
atomInlineOtherContent
+ type: atomMediatype
atomLink
+ href: atomURI
+ hreflang: atomLanguageTag [0..1]
+ length: text [0..1]
+ rel: atomNCName | atomURI [0..1]
+ title: atomText [0..1]
+ type: atomMediaType [0..1]
rel must be either a URI or from the registry of link relations. If not present, alternate is assumed.
atomInlineXHTMLContent
+ type: "xhtml"
Contains text or anyElement
«enumeration»
Link Relations
«enum»
alternate
related
self
enclosure
via
undefinedContent
atomInlineTextContent
+ type: "text" | "html"
Contains an xhtml DIV element
Contains just text content for the parent element.

An excerpt of an example atom feed of Metafor experiments (from a development sandbox) appears as Figure 3.



```
<feed xmlns="http://www.w3.org/2005/Atom" xml:base="http://localhost:8000/">
      <id>http://localhost:8000//feeds/cmip5/experiment/</id>
      <title>CMIP5 model experiment metadata</title>
      <subtitle>Metafor questionnaire - completed experiment documents</subtitle>
      <updated>2010-07-06T00:00:00Z</updated>
      <link href="http://localhost:8000//feeds/cmip5/experiment/" rel="self"/>
      <author>
            <name>The metafor team</name>
      </author>
      <generator version="r33" uri="http://code.google.com/p/django-atompub/"
            >django-atompub</generator>
      <entry>
            <id>urn:uuid:06a22304-8911-11df-bcad-00130286c259</id>
            <title>1.3 noVolcano_10yr_1975 (1.3 hindcast without volcanoes (1975))</title>
            <updated>2010-07-06T00:00:00Z</updated>
            <published>2010-07-06T00:00:00Z</published>
            <link href="/cmip5/experiment/06a22304-8911-11df-bcad-00130286c259/1/"
                  type="application/xml" rel="via"/>
            <summary>Hindcast without volcanoes. Additional 10yr run for experiment 1.1 from 1975
                  without including the Agung, El Chichon and Pinatubo eruptions.</summary>
            <content src="/cmip5/experiment/06a22304-8911-11df-bcad-00130286c259/1/"
                  type="application/xml"/>
      </entry>
```

*Figure 3: Excerpt of the experiment feed that will appear in the production metafor questionnaire.*

It can be seen that the metafor xml payload is visible using both the link and the content mechanism.

## *4 Atom feed infrastructure in METAFOR*

It was determined that it would be easiest to use the Geonetwork system to establish the Metafor "system". Modelling centres would setup Geonetwork servers and expose their content. An aggregator daemon would then auto-populate the CIM Repository. OAI comes "out of the box" with Geonetwork and hence was the initial format for moving content between Metafor locations. However, as noted above, we moved to using the Atom feed capability within Geonetwork and elsewhere.

**Internal Dependencies**

- Metafor is a distributed project, and the document movement infrastructure is integral to the project success, and in particular, the development of the portal.

Figure 4 displays how the various nodes partitioned in terms of those that:

1. create content;
2. consume content (and provide user-services);
3. manage content.

**External Dependencies – Inbound**

- It will be seen that Metafor now depends on an ability to parse and expose information held in ESG data nodes.

**External Dependencies – Outbound**

- Metafor documents are being fed to the Earth System Grid in support of CMIP5 via Atom feeds.
- No services for IS-ENES can be exposed without this activity.
- If we want to interoperate with GENESI-DR we might need the atom feeds (in order to allow them to harvest from us).

Where Geonetwork will be used, i.e. CIM instance editing, the resulting output will be exposed as an Atom feed for ingestion by Metafor Repository services. This has required a small amount of programming to adapt the Geonetwork server to Atom.

### ESG Publisher Integration

Metafor needs to ingest Earth System Grid descriptions of the data held in data nodes. The project has developed a tool which can parse Thredds catalogs to produce CIM data description documents. These then are exposed via an Atom endpoint. Future versions of the cod may use the ESG data node internal PostGres catalog in preference to the Thredds catalog interface.

The latest version of the ongoing work is to be found here:
http://metaforclimate.eu/trac/browser/cimTOOLS/branches/ESG-DB-THREDDS-access

## Questionnaire Atom Feeds

The questionnaire currently exposes: experiments, platforms, models, simulations, and the file endpoint will be supported soon. Tests of the feed access have bene accomplished by the portal devleopment team (internal) and by the ESG team (external).

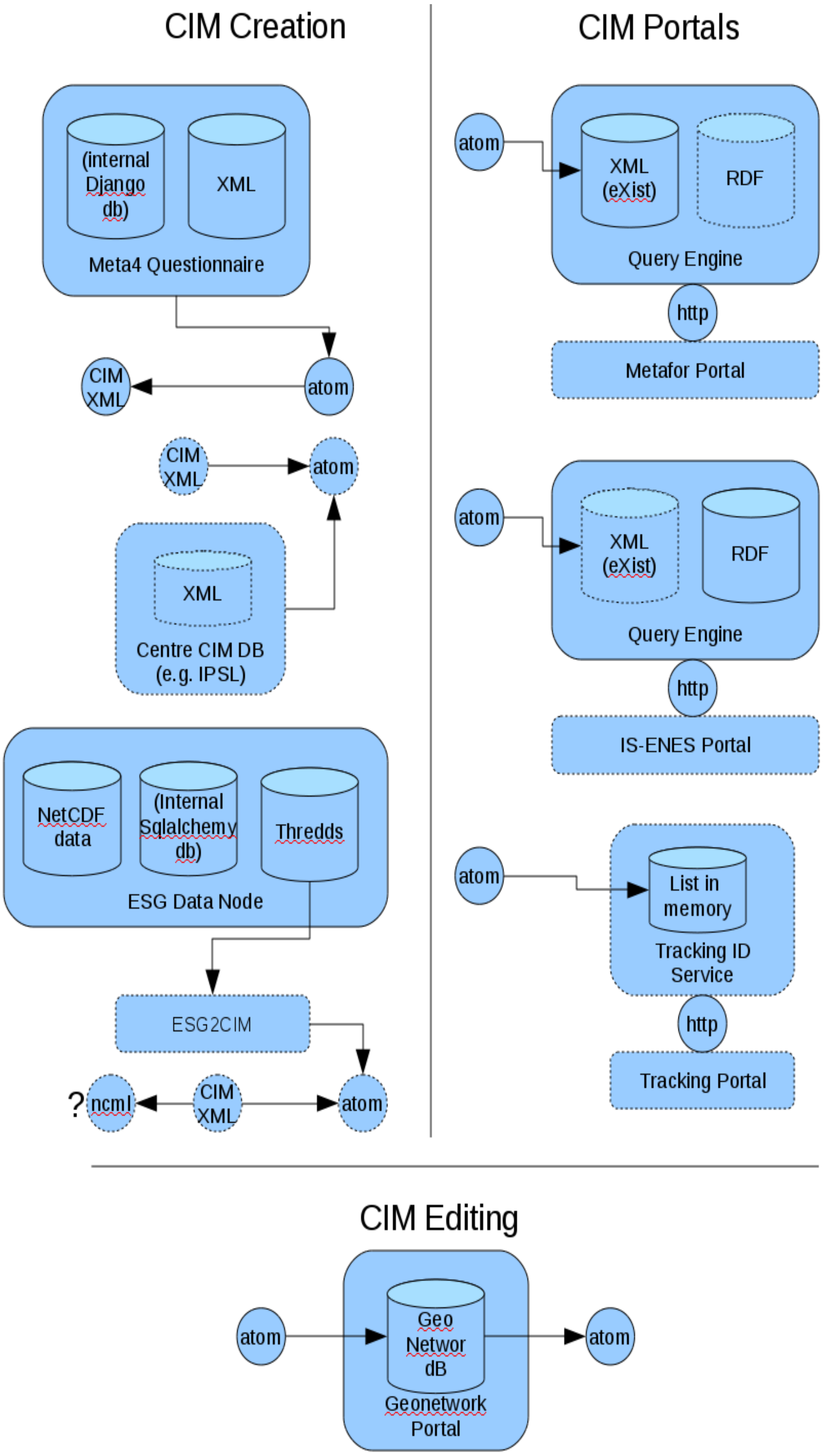


*Figure 4: Key components of the Metafor system: applications which create CIM content (CMIP5 questionnaire, automatic cim generation within modelling infrastructures, and data parsing); applications which exploit CIM content (the metafor and is-enes portals a document tracking service); and applications for editing CIM content (currently the customised geonetwork instance developed in WP6). The dotted outlines indicate components planned, the solid lines indicate existing components.*