METAFOR YEAR 2 REPORT

WP4: Services

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	tbd
Title	METAFOR WP4 Services: Year Two Report
Document Identifier	tbd
Date	March 24, 2010
Work Package	WP4 Services
Authors	BADC, IPSL, UKMO (via AT)
Document Status	Final pending deliverable id and links
Document Link	tbd

Dissemination Level		
PU	Public	X

Document History			
Version	Date	Comment	Author/Partner
0.1	March 15h, 2010	First Draft	C. Pascoe/BADC
0.2	March 19th , 2010	Second Draft	B. Lawrence/BADC
0.3	March 24th, 2010	Final Version	B. Lawrence/BADC

1 INTRODUCTION

An introduction to Metafor appears elsewhere. This document provides, for the Metafor services activity (WP4), a report for the last year (2009-2010) and roadmap to the end of the project – and beyond (as is required for an FW7 I3 project). It has been written as a standalone document so that it can be used as a report to the EC, an internal project document, and a communication to external project partners.

Recall that at the beginning of the project we began with a project plan that had the following expected workflow:

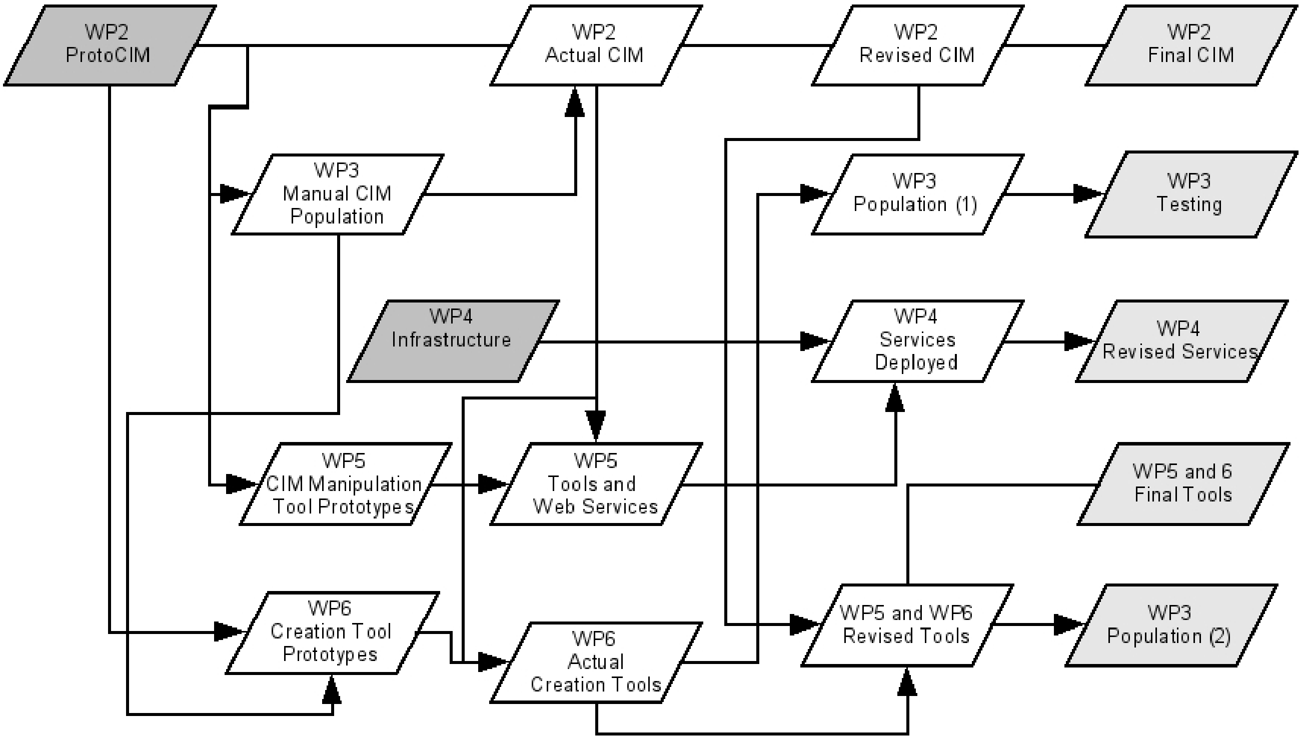


Figure 1: Original Description of Work Pert chart, showing how we expected WP4 to begin by setting up infrastructure, and then exploiting tools (and web services) developed in WP5, to deploy WP4 services in year 3. The initial services were to be deployed using the "Actual" CIM, and then the project was to leave revised services at the project end.

In essence, WP4 was to exploit developments in WP5, themselves exploiting WP2, and deploy services, with "final" versions appearing at the end of the project.

In this document we will lay out how the original vision has been migrated to the current vision, and how that has impacted on the delivery of the milestones and formal deliverables of the project – from a WP4 service perspective. It will be seen that the opportunity of supporting CMIP5 has had significant ramifications, as have developments in other projects. It will also be seen that we have a plan that goes beyond that which is supported by the Metafor funding alone.

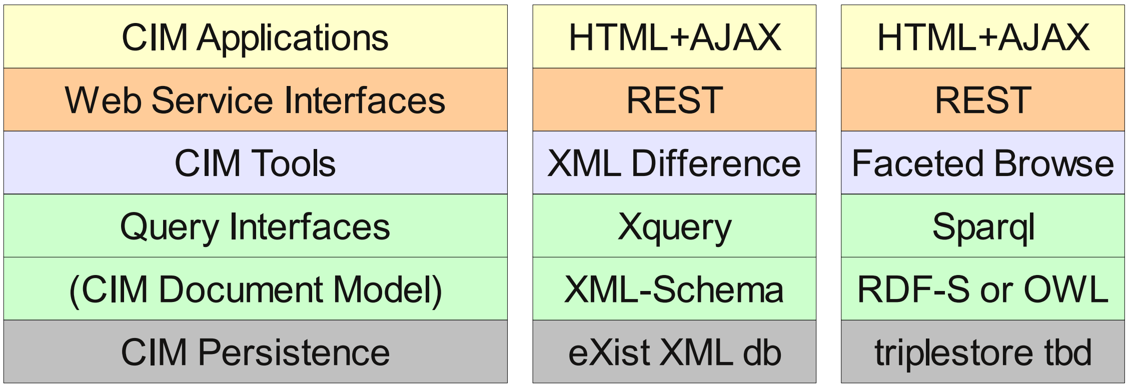
There are two major sections which follow: Progress against objectives (page 3), which provides a brief summary of key activities in the last year along with an appraisal of progress against formal milestones and deliverables; and the Future Roadmap (page 11), which briefly lists some of the issues we need to keep in mind to ensure the project results in activities which have longevity.

2 PROGRESS AGAINST OBJECTIVES

The first stage of the service process was to identify those user services that will be required to fully exploit the CIM (see section 2.5): services to create and update CIM records, to make CIM records available to people, to allow CIM records to be harvested into applications which can exploit the CIM records, services to deploy within those applications, and services to ensure applications can be made aware when CIM records change. In this section we discuss the relationship between services and applications and report on the activities over the last year in support of developing metafor services.

Regrettably throughout this document the word “services” is being used in two distinct and different ways: in the name of the WP, we use the word services to indicate services which are available to users. These “user services” are delivered by web based applications. Within those applications we use “web services” as a key method of delivering functionality. Hence there is obvious scope for confusion whenever we refer to services: as far as possible we try and qualify the noun services with either “user” or “web” to help interpret the material.

Because Metafor is being developed with a layered model (Figure2), the development of user services (portals) and web services (underlying functionality) is completely decoupled, allowing software development in disparate locations at different times against a common architecture.



CIM Applications	HTML+AJAX	HTML+AJAX
Web Service Interfaces	REST	REST
CIM Tools	XML Difference	Faceted Browse
Query Interfaces	Xquery	Sparql
(CIM Document Model)	XML-Schema	RDF-S or OWL
CIM Persistence	eXist XML db	triplestore tbd

Figure 2: Metafor Software Stack: persistence is decoupled from documents, which may be logically arranged in a variety of ways with associated query interfaces. Tools manipulate the query interfaces, and are exposed by web services. Applications exploit the web services to provide user services (generally as HTML pages exploiting Ajax to interact with RESTful web services). Two examples are shown of user services exposed in portals: document differencing (and hence, for example, understanding the difference between two simulations) and faceted browse (which allows sophisticated movement through the complicated network for relationships supported within the CIM). These two examples depend on metafor documents being persisted in different formats, with different syntax (XML versus RDF triples) but a common semantic data/document model (CIM).

Currently, portals with different aims are being developed by three groups:

1. Within Metafor, at BADC, under the auspices of WP4 and WP6, the Metafor CMIP5 questionnaire is effectively a portal which creates and exposes CIM content (using the Django web framework),
2. Within Metafor, at IPSL, the formal Metafor portal for manipulating CIM repositories is being developed using the Pylons web framework, and
3. Within IS-ENES, at DKRZ, a prototype portal which manipulates CIM content is being developed using the Plone framework.

There are compelling reasons why differing frameworks are being used in each case, but the point here is that the metafor architecture supports all three. From a WP4 service perspective then, we need to support both a variety of user and web services.

Figure 3 displays how the various applications are distributed in terms of those which create content, those that consume content (and provide user-services), and those for managing content. It will be seen that Metafor has decided not to deploy OAI/PMH[1] as the main method for moving content (as originally planned): given our software architecture is predicated on RESTful web services the use of the Atom[2] syndication format allows more flexibility (and better performance). To that end, although we have met our OAI prototype deployment milestones, and we could deploy an OAI system, we will instead deliver all the OAI functionality for this project using Atom feeds along with Atom feed parsers (code to both expose and consume Atom is available in most high level languages).

To deliver the applications shown in figure 3, a number of services are required. An early milestone in 2009/10 was the identification of what user and web services were needed to deliver the project. Following that, we've had to consider the appropriate technology stack for each of the user services, and as is clear, we've chosen a range of technologies. While from a purely software engineering standpoint this may not have been ideal, it's both a pragmatic recognition that some things are easier to do in different in technology stacks, and that after the end of Metafor, we need the ongoing support to be done by teams who are familiar with the technology stacks they've deployed.

The primary use case that has been driving Metafor is support for CMIP5. The development and deployment of the questionnaire to *gather* CIM content has both exposed new service requirements and driven team priorities. The Geonetwork developments within WP6 have given us the methodology to *correct* and *edit* CIM content, but it is not a practical method for generating content. Within the next year, the priority will be to *display* and *manipulate* CIM content.

The questionnaire that has been developed will be the major tool for creating CIM content, both for CMIP5, and for other projects. To that end, there have been a large number of questionnaire releases resulting in a “deployed” web-accessible questionnaire during 2009/2010, and the questionnaire has been integrated into the Earth System Grid security framework (itself heavily influenced by the metafor security developments. These three activities: questionnaire deployment and evolution, along with security deployment are discussed below in sections 2.2 to 2.4.

[1] OAI/PMH: The Open Archives Initiative Protocol for Metadata Harvesting, http://www.openarchives.org/pmh/

[2] Atom is a syndication format for XML and other documents, RFC 4287, see http://www.ietf.org/rfc/rfc4287.txt

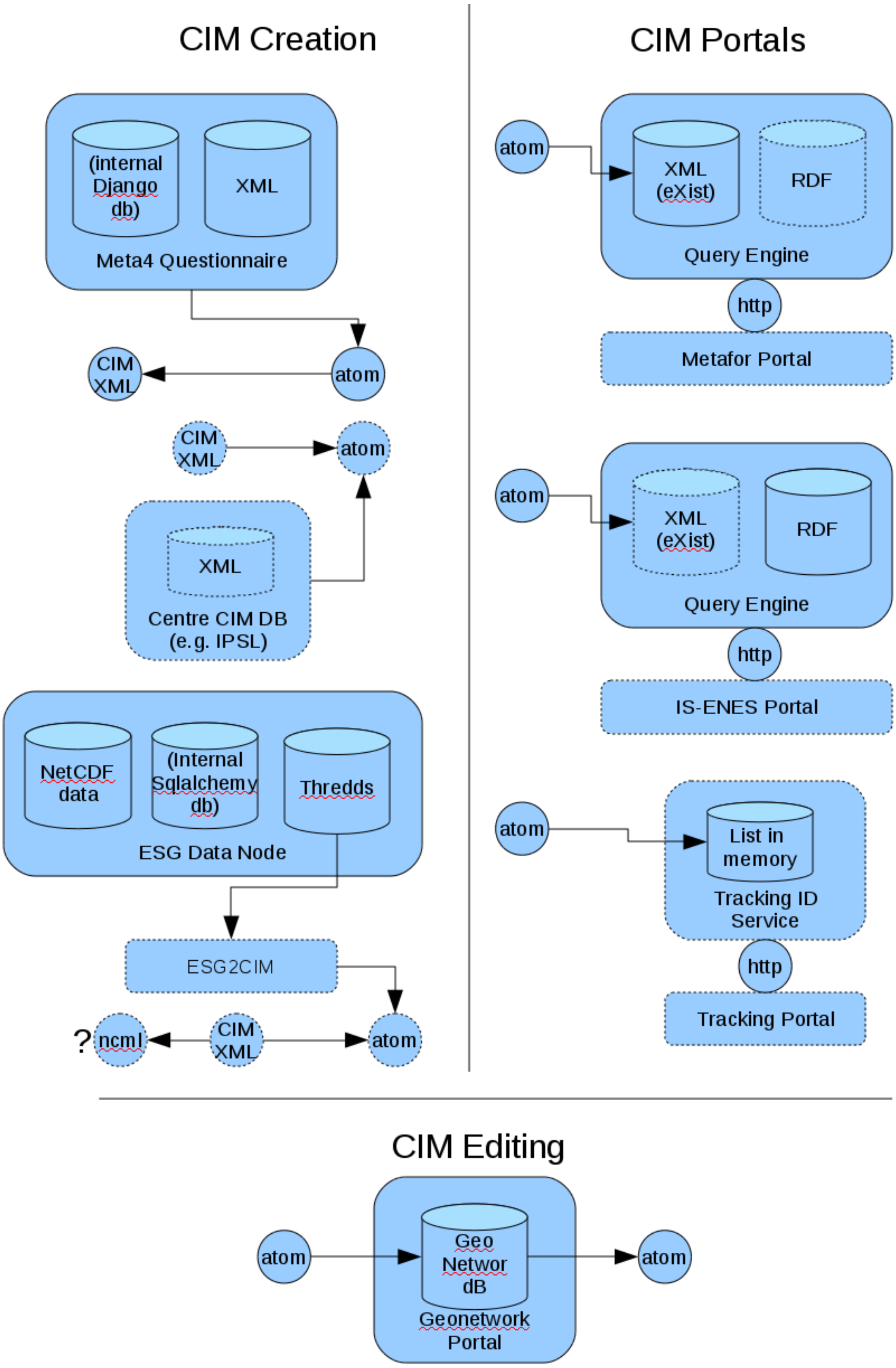


Figure 3: Key components of metafor: applications which create CIM content (the questionnaire, automatic cim generation within modelling infrastructures, and data parsing); applications which exploit CIM content (the metafor and is-enes portals a document tracking service); and applications for editing CIM content (currently the customised geonetwork instance developed in WP6). The dotted outlines indicate components planned, the solid lines indicate existing components.

2.1 ORIGINAL MILESTONES AND DELIVERABLES

These are the original high level milestones and deliverables.

Milestones and Deliverables	Due Date	Status
D4.1 Single-Sign-On Evaluation	M15	**Delivered.** Key result: use OpenID for authentication and SAML for authorisation.
M4.1 Service Identification	M15	**Met**. Note that this service list has been revised. See section 2.5
M4.2 OAI Prototype	M18	**Delivered and Replaced.** OAI systems were deployed, but the project is moving to use of Atom feeds for document exposure and harvest. See section 2
D4.2 OAI System Deployed	M24	**Slight Delay (circa 2 months).** Replaced with deployed atom system. Atom systems have been deployed as originally envisaged for OAI, however new software is now required for interface to the Earth System Grid.
M4.3 Portal Prototype	M27	**On target.** (Initial portal prototypes were delivered in year one) A new development is now underway exploiting pylons and ajax technology to expose web services developed in and the first version of that will
M4.4 Secure Services Deployed	M27	**Already Achieved.** Authentication was required for the Metafor CMIP5 questionnaire, and is now deployed. Authorisation is not necessary at this stage for Metafor, but is under testing anyway.
M4.6 Data Transformation Tools Deployed.	M27	**Likely to be delayed.** A number of transformation tools will be deployed, but it is likely a more complete set will be deployed around M30.
D4.2 Portal Deployed	M30	**On target, but delay anticipated.** This will be necessary for the CMIP5 project, as we anticipate that DOI's will land on the Metafor portal. It is possible that the consequential CMIP5 requirements may cause a slight delay & not all services will initially be deployed in the portal. However, a new and unexpected service, the Metafor Questionnaire will be deployed in advance of the formal portal, and effort has been targeted on that development as a priority.
D4.4 Help Desk	M30	**To be met early.** When the Metafor metadata questionnaire is deployed (expected April 2010), help desk services will be needed. On target.
D4.5 Service Report and Revised Infrastructure	M36	**On target.**

2.2 CMIP5 QUESTIONNAIRE DEPLOYMENT

The CMIP5 Questionnaire has been developed using the Python web development framework Django. The application itself is run in a WSGI container. WSGI (the Python Web Server Gateway Interface) provides a convenient building block for developing web applications and middleware. In this case, it provides two distinct advantages:

- It can be hosted in the Apache web server using mod_wsgi a robust and scalable execution environment for running Python web applications.
- As a WSGI component, the questionnaire application can be combined with security WSGI middleware components to overlay access control functionality. WSGI components are arranged in a pipeline ending in the Questionnaire application. Requests are intercepted by each security middleware component in turn filtering the request or passing to the next middleware in the pipeline. If the request is granted, it passes through to the Questionnaire application to serve a response.

In addition the application exploits a POSTGRES database running on a production database server within the BADC to store incomplete user generated content. Completed content is stored as XML files and made available using the Atom feed.

The Questionnaire and security WSGI components are released as Python Eggs. A Python egg encapsulates a distribution for a given Python package. When an installation is required on a given host, the required eggs may be pulled from egg repositories where they are held. This task is managed using zc.buildout a system for building, assembling and deploying Python applications. Together with a zc.buildout configuration, an INI file sets the configuration of the WSGI pipeline and any other static settings needed such as database connection and security parameters.

The Questionnaire is hosted on a Xen based Virtual Image providing both a flexible means for deployment, and more reliability: failure of a physical server can be recovered quickly by moving the Xen image to a new physical server relatively quickly.

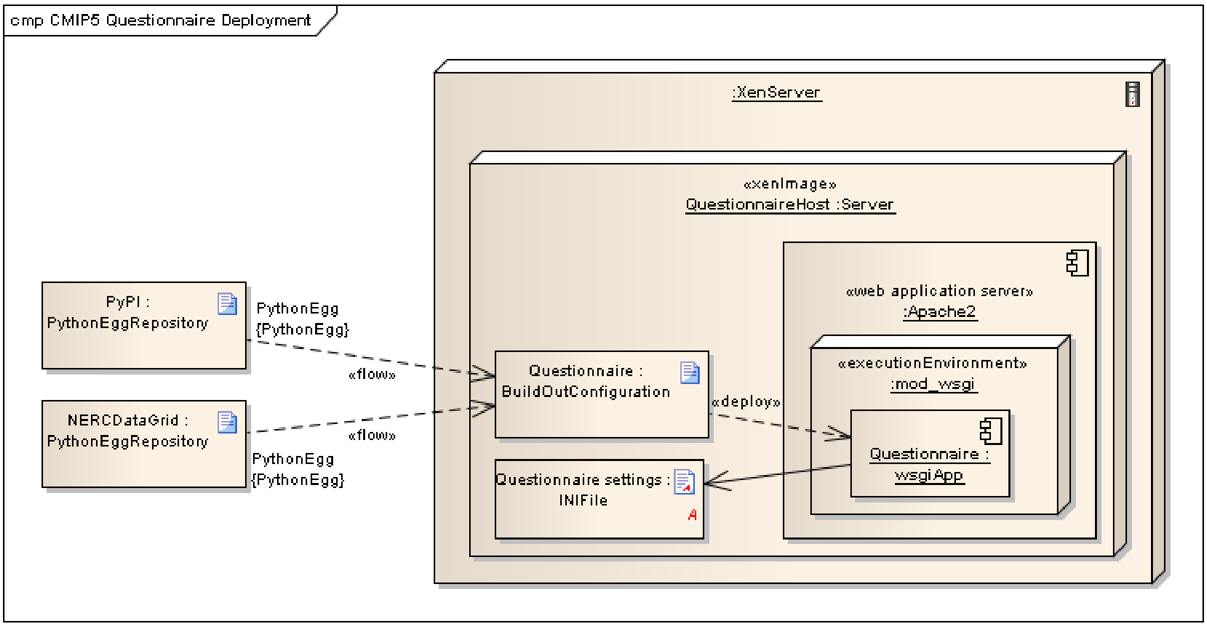


Figure 4: Questionnaire deployment using Apache2 and mod_wsgi within a Xen image. The links to the database server are not shown in this schematic.

2.3 EVOLUTION OF QUESTIONNAIRE

The development paradigm for the CMIP5 questionnaire required that a working version be available at all times. The questionnaire has evolved through numerous iterations as bugs have been found and fixed and the functionality has been extended and improved following feedbac. Initially the regular Alpha deployments of the questionnaire were released only to the Metafor community. In late 2009 the first Beta deployment was made available to the wider atmospheric community, the iteration cycle of Beta deployments occurred on a longer timeframe than the Alpha deployments and substantial beta upgrades of the questionnaire were given new URLs. The long iteration cycle and the persistence of old versions of the questionnaire ensured that the beta testers were offered a stable product in which to test their model descriptions. The schedule of questionnaire deployments are summarised in table 1 below.

Version	URL http://	Notes	Release Date
Alpha-1	cmip5.metafor.ceda.ac.uk/cmip5	Prototype questionnaire released to Metafor partners	2009/07/27
Alpha-2	cmip5.metafor.ceda.ac.uk/cmip5	Realised that coupling should be made a priority for next release	2009/08/03
Alpha-3	cmip5.metafor.ceda.ac.uk/cmip5	Support for data objects. Support for saving conformances. Initial support for coupling	2009/08/12
Alpha-4	cmip5.metafor.ceda.ac.uk/cmip5	Included the whole coupling cycle	2009/09/10
Alpha-5	cmip5.metafor.ceda.ac.uk/cmip5	Secured with ESG/Meta4/NDG security	2009/09/29
Alpha-6	cmip5.metafor.ceda.ac.uk/cmip5		
Alpha-7	cmip5.metafor.ceda.ac.uk/cmip5	Implemented bundling of model components	2009/10/23
Alpha-8	cmip5.metafor.ceda.ac.uk/cmip5	The first examples of preloaded files are made available.	2009/10/30
Alpha-9	cmip5.metafor.ceda.ac.uk/cmip5	mindmap updates	2009/11/06
Alpha-10	cmip5.metafor.ceda.ac.uk/cmip5	Aerosols and Atmospheric Chemistry were split into separate realm components. The questionnaire crashes “gracefully”	2009/11/18
Beta-1	cmip5.metafor.ceda.ac.uk/cmip5	Prototype questionnaire for wider release	2009/11/30
Beta-2	ceda.ac.uk/cmip5	Open-id security implemented	2010/02/08
Beta-3	ceda.ac.uk/cmip5	Bug fixes	2010/02/19
Beta-4	q.cmip5.ceda.ac.uk	Tick boxes on drop down lists Prototype grid support	2010/03/08

2.4 SECURITY INFRASTRUCTURE

Metafor has exploited an evolution of the NDG (NERC DataGrid) Security software which provides access control implemented in the Python programming language. It, like the questionnaire, exploits the Python WSGI standard to realize a flexible component based architecture based on WSGI middleware building blocks. These components are independent of the application they secure so that potentially any existing Python based HTTP application may be secured using the middleware. Individual WSGI components define different aspects of the access control functionality, enabling custom combinations of middleware to be assembled together to meet a given set of access control requirements.

The system has been developed alongside an the US Earth System Grid project for the purposes of ensuring a single identify management system for metafor and CMIP5 (since metafor systems are integral for CMIP5 metadata handling). The adoption of widely accepted standards such as SAML and OpenID has ensured interoperability with the ESG Java based software implementation. The filter based architecture has also been adopted for the Java implementation with individual middleware components defined as Java Servlets.

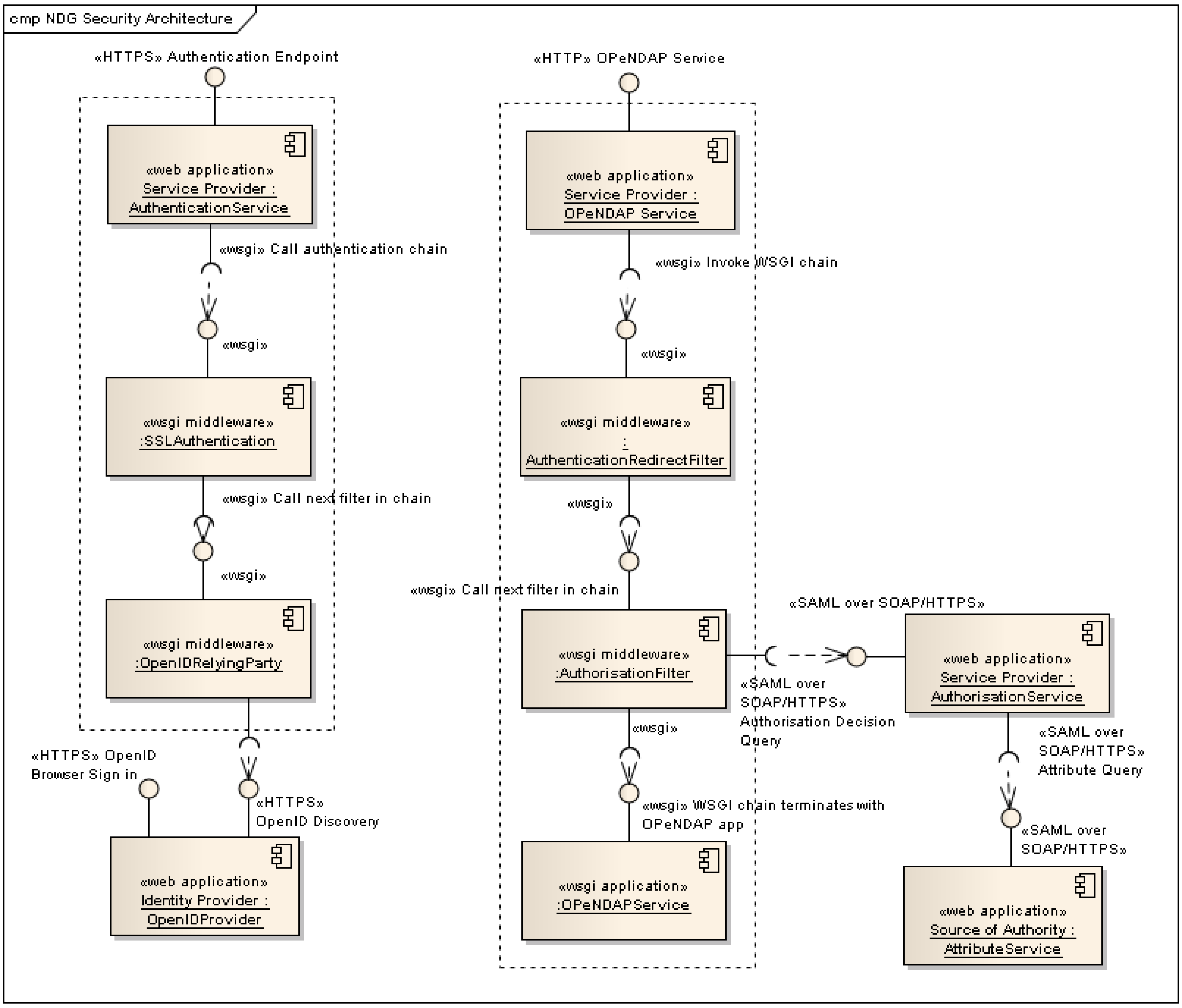

cmp NDG Security Architecture
«HTTPS» Authentication Endpoint
«web application» Service Provider : AuthenticationService
«wsgi» Call authentication chain
«wsgi»
«wsgi middleware» :SSLAuthentication
«wsgi» Call next filter in chain
«wsgi»
«wsgi middleware» :OpenIDRelyingParty
«HTTPS» OpenID Browser Sign in
«HTTPS» OpenID Discovery
«web application» Identity Provider : OpenIDProvider
«HTTP» OPeNDAP Service
«web application» Service Provider : OPeNDAP Service
«wsgi» Invoke WSGI chain
«wsgi»
«wsgi middleware» : AuthenticationRedirectFilter
«wsgi»
«wsgi» Call next filter in chain
«wsgi middleware» :AuthorisationFilter
«wsgi»
«wsgi» WSGI chain terminates with OPeNDAP app
«wsgi application» :OPeNDAPService
«SAML over SOAP/HTTPS»
«SAML over SOAP/HTTPS» Authorisation Decision Query
«web application» Service Provider : AuthorisationService
«SAML over SOAP/HTTPS» Attribute Query
«SAML over SOAP/HTTPS»
«web application» Source of Authority : AttributeService

Figure 5: Security architecture: In the middle we see a WSGI middleware stack making up the configuration for a secured application. In this case it is an OPeNDAP service shown at the bottom but it could equally be any other WSGI based Python applicatio such as the CMIP5 Questionnaire.

2.5 SERVICE IDENTIFICATION

As at milestones 4.1: At the formal milestone, we identified the following service requirements, not all of which might be achievable within Metafor, and we formally noted the expectation that we might change/add to this list as development proceeded.

User Services:

1. Atom feed of new CIM instances at all sites producing CIM content for wider accessibility (some CIM content will of course be only of local interest).
2. An aggregation service – to harvest CIM compliant metadata from CIM producers.
   - In practice this will be achieved by exploiting Atom feed parsers in any applications exploiting the CIM. We may need a central registry of CIM production feeds.
3. An generic search interface (possibly including OpenSearch as well)
   - To be deployed in the metafor portal.
4. An annotation interface for CIM instances
   - To be deployed in the metafor portal, possibly also in the questionnaire for quality control management by data and metadata producers.
   - Along with a feed or feeds of such annotations
5. An editing service for CIM instances
   - (being delivered by the geonetwork developments)
6. A WFS interface to CIM content
   - This is unlikely to be delivered within Metafor as there are no internal use case drivers, but we can see that eventually interoperability requirements may make this useful.
7. A service to allow differencing CIM documents so users can understand difference in simulations, experiments, models and data.
   - To be deployed in the portal exploiting the web services developed in WP5
8. A service to expose underling CIM transformation tools (such as validation, pretty printing, xml to rdf conversion, and vice versa, etc).
9. Query interfaces exposing the underlying web services so organisations can develop their own portals etc exploiting harvested CIM content.
   - This may be one of the methods of allow IS-ENES portals and Metafor portals to exploit co-development. See section 3.1.
10. Vocabulary services exploiting a RESTful API – including editing services to allow the management of controlled vocabularies.
11. Governance services allowing the upload of new version of the ConCIM UML and the download of conformant ApCIM XML schema and (hopefully, see section) RDF-S or OWL schema.

3 FUTURE ROADMAP

Clearly in the next year, Metafor both needs to meet it's contractual obligations and put in place systems that ensure the ongoing support and development of the ideas, software and services arising from the project.

One obvious driver will be the necessity to continue to work with the Earth System Curator and Earth System Grid teams to ensure the availability of CMIP5 metadata tools throughout CMIP5. Similarly we will need to work with the funders of the IPCC activities to ensure that those Metafor tools which will become entrained in IPCC systems can be persisted long term.

Within the services stream, the key activities to manage are:

1. Establishing services to support the governance of metafor infrastructure: in particular, to support managing the evolution of the CIM and the software built upon it (to be clear,
2. The ongoing deployment of the CMIP5 questionnaire (which will need to be deployed for the duration of the CMIP5 project, expected to be many years), and
3. The Metafor portal, which will need to be deployed until at least mid 2012 (the latest cut off date for data to be included as citations in the next IPCC assessment report).
4. Vocabulary services so that vocabularies developed within metafor can transition from ad hoc management by team members into formal web services which can be governed by appropriate communities of scientists, but delivered reliably.
5. Transitioning the key software components into open source projects that can be maintained by the community (with or without formal funding lines), and
6. The establishment of a clear URI and resource naming conventions that will provide persistent access to all distributed CIM based resources for the foreseeable future, and
7. The delivery of help desk services for all of the above.

Clearly one of the projects which will be exploiting Metafor developments will be the InfraStructure for a European Network for Earth Simulation (IS-ENES). While Metafor expects to deliver against all the objectives of the original project plan, we need plans for the ongoing support of user services etc (as described in the introduction above). In addition, the community has clearly moved on: both in terms of societies expectations of what should be documented and available, and in terms of the technical possibilities associated with delivering commensurate services.

To that end, metafor and is-enes met recently and established a clear plan for

1. Which activities should be completed within metafor,
2. Which could be picked up by IS-ENES,

We have yet to clearly identify

3. Which activities need to be picked up by national funding, and
4. Which activities need to be picked up at the European Level

The metafor final report will address these four categories in detail, the following section covers only the first two, and only from a user and web service perspective.

3.1 CO-EVOLUTION WITH IS-ENES

One of the clear areas where metafor and is-enes are collaborating is in the development of differing services which exploit CIM content. Metafor is concentrating on services which exploit an (XML) document based view of content, and is-enes is exploiting an (rdf) graph based view of content. Two differing portals are currently being built as outlined in Figure 3 - exploiting the layered nature of the metafor architecture (Figure 2).

Moving forward we clearly need to coordinate and exploit developments within each project. The following list is included here as a guide to how we plan these developments, and can be used as a guide to understand how we plan to meet our formal metafor project deliverables in year 3. (Here we use the familiar terms "ConCIM" to denote the UML Conceptual Model for the Metafor Common Information model, and "ApCIM" to denote Application Schema in a variety of serialisation formats.)

1. Metafor needs to promulgate a clearly understood URN/URI naming scheme so that documents generated anywhere can be replicated and broken into pieces, with clear understanding when two documents or fragments are identical (but replicated) or referring to different objects or different versions of descriptions of the same objects.
2. Metafor will release 1.4 CIM XSD in early April (with the formal release of the questionnaire). (From a service perspective this version will correspond to the "Revised CIM" in Figure 1).
3. WP5 will build services based on xquery exploiting 1.4 instances persisted as XML documents.
4. WP4, as part of developing support services for the CIM, will help design the migration of the metafor UML "meta-model" (that is, the set of rules we use to construct the "ConCIM" - the UML CIM structure) to one that is consistent with standards compliant ApCIM XML schema generation (in particular, the HollowWorld[3] and FullMoon[4] formalism). Additionally,
5. We will also examine the prospects of generating RDF-S or OWL versions of the ApCIM, so that
6. Ideally Metafor can leave CIM 2.0 in such a way that
   - The V2.0 ConCIM is HollowWorld and FullMoon compliant (so future developments in the ConCIM can be easily translated to XML schema ApCIM),
   - We either understand how to, or can actually, automatically generate the ApCIM in RDF-S or OWL as well, and
   - IS-ENES will be in a position to automatically decompose ApCIM compliant XML instances into RDF triples, and
   - Vocabularies have been decoupled from the CIM in such a way that instances can be validated both against schema and the vocabularies.
7. In parallel, IS-ENES will be investigating RDF tooling, initially exploiting RDF instances harvested via OAI-PMH from ESG gateways (thus exploiting the bespoke XML to RDF triples code jointly developed by Earth System Curator and Metafor projects)
8. IS-ENES will then migrate to RDF triples conforming to CIM-2.X-RDF, when it is available
   - (which could be semantically different from the 1.4 being used for ESG and CMIP5).

[3] HollowWorld: see https://www.seegrid.csiro.au/twiki/bin/view/AppSchemas/HollowWorld

[4] FullMoon: see http://projects.arcs.org.au/trac/fullmoon/

9. The metafor portal will concentrate on exploiting xquery tooling and supporting CMIP5 citation infrastructure. It will be relatively static from early 2011, although it might be able to exploit backported RDF to add functionality from the IS-ENES development or direct engagement with web services exposed by IS-ENES (This possibility is indicated by the dotted lines around the RDF persistence for the metafor portal shown in Figure 3).

10. The IS-ENES portal will continue to evolve, and have enhanced functionality. Backend services may include an evolved version of the metafor xquery infrastructure consistent with CIM 2.0 either implemented locally or as web service calls to the metafor portal (This possibility is indicated by the dotted lines around the XML persistence for the is-enes portal shown in Figure 3).

11. Metafor will develop and deploy a standalone tracing ID service, which takes a “tracking-id” from a CMIP5 data file, and provides a URL to metafor, esg, and is-enes portal representations of the associated metadata).