DRIHM

DISTRIBUTED RESEARCH INFRASTRUCTURE FOR HYDRO-METEOROLOGY

Report on the Inventory of Deployed Services

Abstract: This document lists the inventory of deployed DRIHM services based on a given architectural model of the HMR e-infrastructure.

DRIHM (G.A. n° 283568) is co-Funded by the EC under 7th Framework Programme

Document Information Page

Contract Number	283568
Project Name	Distributed Research Infrastructure for Hydro-Meteorology
Project Acronym	DRIHM
Deliverable Number	D5.2
Deliverable Name	Report on the Inventory of Deployed Services
Work Package Number	5
Work Package Name	Infrastructure Operation and Management, Data and Resource Management
Deadline	2012-03-01
Version	1
Dissemination Level	Public
Nature	Report
Lead Beneficiary	LMU

Table of Contents

1 Executive Summary

This deliverable is a living document. It serves as a report on the inventory of deployed and exploited middleware services in the context of the DRIHM project. The initial version of this report covers the state-of-affairs by month 6 of the project. Before month 12 the report will be reviewed as explained below. Subsequent versions will provide addenda for months 18, 30 and 42.

In principle, most of the middleware and application services necessary for the DRIHM project are expected to extend existing capabilities as already provided by European e-infrastructures, thus ensuring DRIHM interoperability and integration. At the same time the main objective of newly deployed services is to meet specific requirements stemming from DRIHM applications – mainly the experiment suites – as determined in [1]. For these reasons, in DRIHM, the service deployment process is iterative and will span most of the project lifetime.

In this deliverable we describe the services relevant for DRIHM as they are available at the time of writing. Considering that the first release is due at a very early stage of the project (month six) it describes the DRIHM plan for service selection and deployment rather than any actual deployments. As soon as the first set of services will be deployed this version of the report will be updated with an initial list of services. After that, new versions of this deliverable will extend the list both in terms of the number of middleware services and in terms of their integration with middleware services provided over European e-infrastructures like EGI and PRACE. Additionally, future versions will also update the number of resource providers supporting the respective middleware services and tools.

This report must be considered as a complementary document to reports D5.1 (Report on the Assessment of Operational Procedures and Definition of the DRIHM Operational Model) and D5.4 (Report on Support Process Definition).

2 Introduction

One of the main goals of the DRIHM project is the implementation and operation of a distributed and inter-organisational IT-infrastructure for hydro-meteorologic research (HMR) capable of supporting the execution of complex HMR workflows like those described in the DRIHM experiment suites mentioned in [1]. In order to successfully fulfill the objectives related to the DRIHM goals, DRIHM users need to rely on adequate services on different layers of an envisioned HMR service stack [2]. Consequently, the focus of WP5 is on the adaptation and the deployment of such services, whether they are existing ones or new ones to close gaps stemming from the specific requirements of the DRIHM service infrastructure, to be determined by work packages 6, 7 and 8 as depicted in Figure 1. It is important to notice that at the time of writing this first version of the report (month 6), WP6 has not yet started (it will start in month 8). On the other hand WP6 will define the handover from service design-selection and early implementation to service deployment, thus representing the actual connection between WP5 and research activities carried out in WP7 and WP8 [1].

Figure 22 conceptually depicts the DRIHM layered architecture. The service inventory described in this deliverable will progressively (that is in future versions) relate to all layers and will list the respective services deployed for DRIHM. Looking at the DRIHM layered architecture, as depicted in Figure 2, it is evident that services at the lower architectural levels are expected to be less sensitive to HMR requirements. Consequently, they are going to be deployed before high level services. However, exceptions may be possible.

Note that DRIHM aims at infrastructure integration and at interoperability with existing middleware services. For these reasons, DRIHM leverages services already provided by existing e-infrastructures and middleware technologies, "DRIHMifies" them where appropriate, and orchestrates them for HMR specific workflows designing and implementing new services when required. Pursuing integration and interoperability requires an in-depth comparative analysis with respect to existing infrastructures and middleware currently available or being developed in the European Grid ecosystem.

Considering the situation described above, the current state of affairs as far as service deployments are concerned (at month six), is discussed in this report. The main aspects are:

- An analysis of available middleware technologies and services. In particular DRIHM has strong contacts with:
  - EGI in order to setup a HM Virtual Research Community in its framework;
  - MAPPER project (http://www.mapper-project.eu/). This project is of great interest because it pursues integration of multi-scale simulation models.
- The cooperation with WP 7 and WP 8. Waiting for the official start of WP6 it has been possible to grasp useful indications from WP7 and WP8 in particular through the activity concerning the following aspects:
  - On the Service Layer DRIHM will rely on several models. The metadata of these models have been collected in the FluidEarth catalogue (http://catalogue.fluidearth.net/) [5][1]. For the most relevant models we will just repeat these metadata here.
  - DRIHM also uses events described in the HyMeX Database (http://mistrals.sedoo.fr/HyMeX)[1] [6] . However, the HyMeX events are rather dynamic and thus not foreseeable. Instead of describing in this document all events, we therefore refer to the HyMeX database.

This document will be updated regularly to reflect the current state of the DRIHM project and of the European research infrastructures.

[1] Registration required

Context

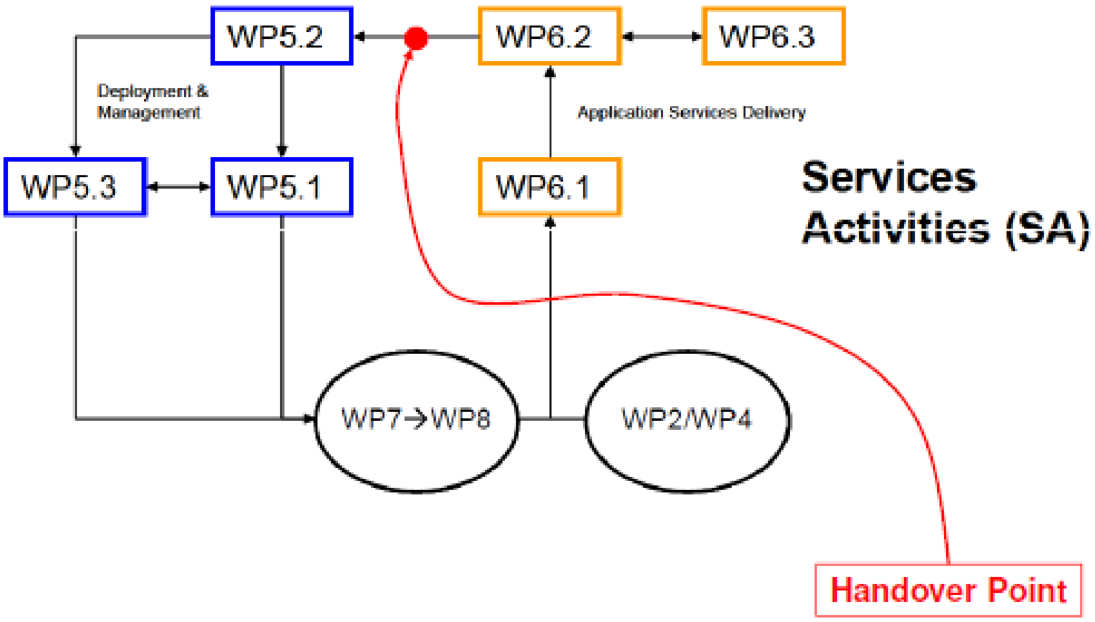

Figure 1: DRIHM Service design and implementation process

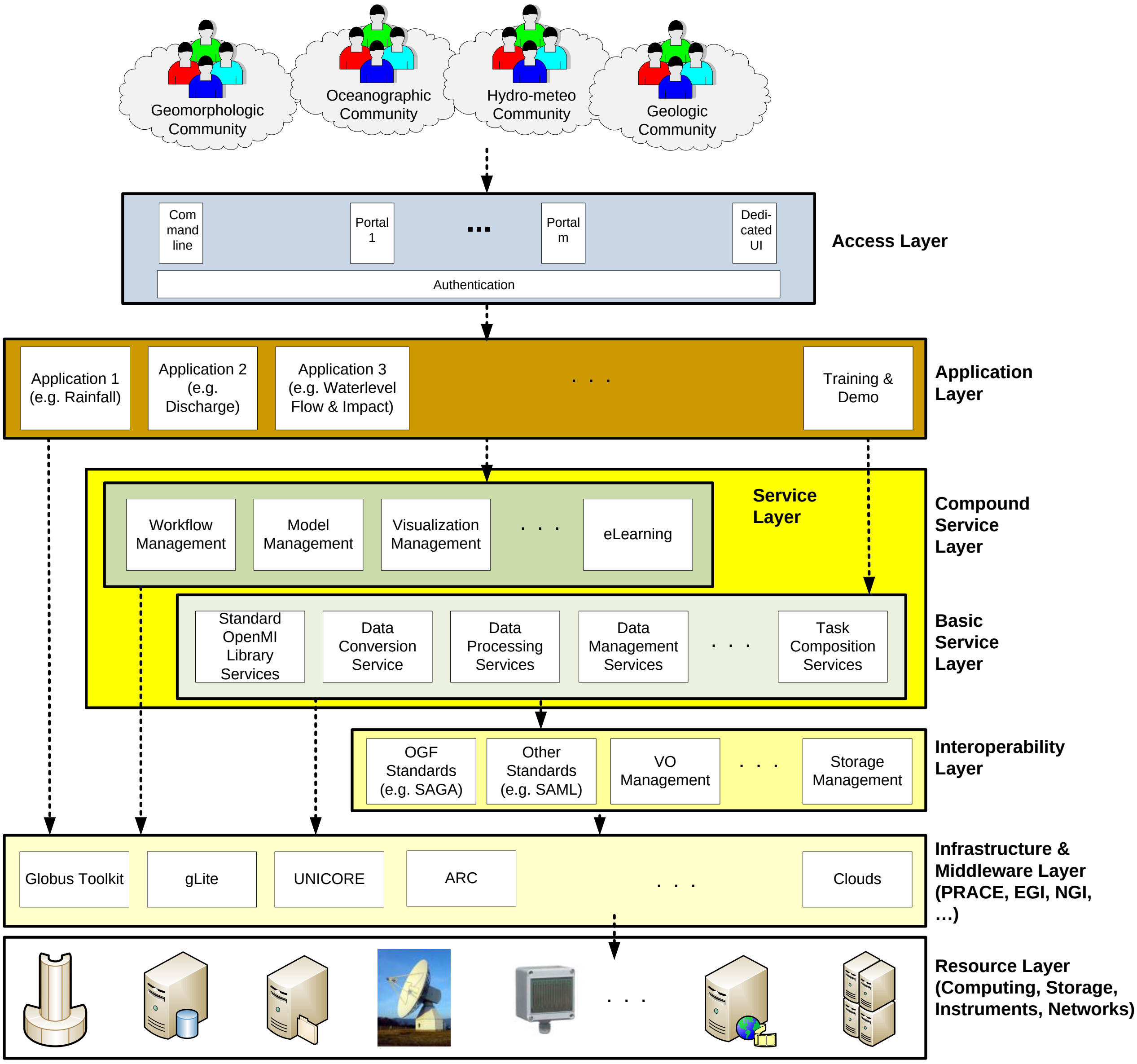

Figure 2: DRIHM layered architecture

3 Selection Criteria

This section describes the main criteria for selecting services to be included into the DRIHM architectural stack.

As already mentioned in [1] and [2], the (technical) objective of the DRIHM project is to provide a set of services to be deployed over several European e-infrastructures (e.g., PRACE) and to be used for HMR scientists (including so called citizen scientists) for solving HMR related problems like those exemplified by the DRIHM experiment suites [1,2].

Consequently, DRIHM has to cope with heterogeneity on all layers of the stack depicted in Figure 2: heterogeneous resources, heterogeneous Grid middleware, heterogeneous services, heterogeneous models, and even heterogeneous access policies. This heterogeneity, though, is not by accident. Rather, it is imposed first by the European e-infrastructure and their Grid implementations; and second by the nationally preferred models and their access / usage policies.

The selection criteria have to reflect this.

3.1 Selecting Services for the Resource, Infrastructure and Middleware Layers

Due to its close relationship with DEISA/PRACE, EGI and the various national Grid initiatives (NGI) (see also [4]), the initial version of this deliverable will inventory services and tools related to accessing these infrastructures. Future versions of this document will additionally reflect (see also Figure 2) NorduGrids Advance Resource Connector (ARC)[2] and Cloud

[2] http://www.nordugrid.org/arc/

infrastructures (e.g., Science Clouds (http://scienceclouds.org/), Venus-C (http://www.venus-c.eu/Pages/Static/Mission.aspx), or the Amazon Elastic Compute Cloud (http://aws.amazon.com/de/ec2/)).

3.2 Selecting Services for the Interoperability Layer

Technically speaking, the Interoperability Layer refers to services which in a standardized way hide the heterogeneity inherently associated with the underlying Resource, Middleware and Infrastructure Layers. Since DRIHM will leverage Grid infrastructures, the services and tools on this layer need to allow for a standardized access to Grid infrastructures. Consequently, this layer contains services and tools for the management of Virtual Organizations (VO), for accessing Grid resources (Computing Elements, Storage Elements, Instrument Elements), for expressing authorization and entitlement policies (like OASIS' XACML (http://www.oasis-open.org/committees/xacml/charter.php)), for security management (like OASIS' SAML (http://www.oasis-open.org/committees/security/charter.php)), and for mastering the compliance of DRIHM applications to OGF application standards like SAGA (http://www.gridforum.org/gf/group_info/view.php?group=saga-wg) or DRMAA (http://www.gridforum.org/gf/group_info/view.php?group=drmaa-wg). Note that DRIHM will not provide any interoperability service per se. Instead, DRIHM will commit itself to apply existing standards and adhere to interoperability services already deployed.

Note that the inherent DRIHM model interoperability is part of higher levels as this interoperability issue is not directly related to Grid infrastructures.

3.3 Selecting Services for the Service Layer

The Service Layer is the central area for the DRIHM project as it provides the DRIHM central services to be deployed across the European e-infrastructures and to be used by the HMR communities (including the citizen scientists). Obviously, on the service layer basic services can be separated from more complex ones. While the former are conceptually contained in the Basic Service Layer, the latter ones comprise the Compound Service Layer (see Figure 2).

The Basic Service Layer provides all services necessary for the management of event data (see e.g., the HyMeX database [6]). Example services are format conversion, model transformation, metadata comparison, or OpenMI library services [7]. OpenMI services are necessary since DRIHM has decided to consider OpenMI one of the main model wrappers [1].

The Compound Service Layer uses services provided by the Basic Service Layer to provide more complex DRIHM services like workflow management, model management, visualization or (later) e-Learning capabilities. Since a basic DRIHM requirement is the seamless composition of models, adequate tools need to be considered for this layer.

3.4 Selecting Services for the Application Layer

The Application Layer contains the DRIHM experiment suites (considered as services) [1] and training/demo applications as required. The experiment suites will be developed by work packages 6, 7, 8. For further information we refer to the respective reports to be issued by WP6, WP7, WP8.

3.5 Selecting Services for the Access Layer

The Access Layer provides services to access (read, write, execute) Application Layer services either by using command line interfaces or portals or proprietary user interfaces. All services on this layer are required to authenticate users. DRIHM is currently evaluating so called Science Gateways (https://portal.xsede.org/science-gateways).

4 Descriptions of Services Relevant for DRIHM

In the following we will briefly describe per layer the services and tools necessary for a successful DRIHM deployment.

4.1 Resource Layer, Infrastructure and Middleware Layer, Interoperability Layer

The main focus of the Resource, Infrastructure and Middleware Layers is the transparent use of existing (and future) European e-infrastructures today exemplified by DEISA, EGI, PRACE and others. These infrastructures can be classified today by

- Mainly Unicore-based (like DEISA/PRACE)
- Mainly gLite-based (like EGI)
- Mainly Globus Toolkit-based (like many of the NGIs)

Typically, these infrastructures are associated with a set of local resource management systems like Torque (http://www.adaptivecomputing.com/products/torque.php), the GridEngine (http://gridscheduler.sourceforge.net/), or Platform Computing's LSF (http://www.platform.com/workload-management/high-performance-computing). We also find on top of them dedicated Grid middleware access layer components like the Unicore 6 implementation of the OGSA[3] – Basic Execution Services (BES) standard or the gLite CREAM[4]

[3] OGSA: Open Grid Services Architecture, an OGF standard, Unicore BES User Manual: http://www.unicore.eu/community/development/OGSA-BES/UNICORE_OGSA_BES_UserManual_Edition_1_0.pdf

[4] http://grid.pd.infn.it/cream/

(Computing Resource Execution And Management) service for job management operations at the gLite Computing Element (CE) level. The Globus Toolkit middleware implementation already includes an implementation of an access layer which, however, is not OGSA-BES compatible.

For accessing and using Grid resources in a more transparent manner, the EU-funded MAPPER project (http://www.mapper-project.eu/) defined in [3] a set of additional services, DRIHM will use as well. The following discussions are directly adapted from [3]:

- **HARC**
  HARC, the Highly-Available Resource Co-allocator [9] developed at the Louisiana State University is an extensible, open-source system for creating and managing resource reservations. HARC uses Paxos Commit [10] protocol to reserve multiple resources in a single, indivisible step. The most common use of HARC is to make advance reservations on multiple supercomputers. This functionality will be used by DRIHM and the QCG-Broker (see below) to co-allocate NGI resources for running cross-cluster applications like DRIHM workflows by the AHE service (see above).
- **QosCosGrid**
  QosCosGrid [12] was designed as a multilayered architecture being capable of dealing with computationally intensive large-scale, complex and parallel simulations that are often impossible to run within one computing cluster. The QosCosGrid middleware enables computing resources (at the processor core level) from different administrative domains to be virtually welded via Internet into a single powerful computing resource. QosCosGrid delivers a ready-to-use stack of Grid middleware software tightly integrated with commonly used programming and execution environments for large scale parallel simulations, such as OpenMPI or ProActive [13]. Supporting a wide range of development frameworks as well as programming models relevant for DRIHM application

developers, QosCosGrid gives the ability to work across heterogeneous computing sites hiding the complexity of the underlying e-infrastructures by simplifying many complex deployment and access procedures. QosCosGrid services extend the functionality provided by the gLite and Unicore infrastructures offering advance reservation capabilities needed to co-allocate various types of resources required by the DRIHM experiment suites.

- **QCG-BES/AR**
  The QCG-BES/AR [12] service (also known as the Smoa Computing) is an open architecture implementation of SOAP Web Services for multi-user access and policy-based job control routines by various Distributed Resource Management systems. It uses DRMAA (see section 3.2) to communicate with the underlying resource management systems. QCG-BES/AR has been designed and implemented in the way to support different plugins and modules for external communication. Consequently, it can be used and integrated with various authentication, authorization and accounting infrastructures and other external services. QCG-BES/AR service is compliant with the OGF HPC Basic Profile [14] specification, which serves as a profile over the Job Submission Description Language (JSDL) [15] and OGSA BES standard. In addition, it offers remote interface for Advance Reservations management, and support for basic file transfer mechanisms. The service was successfully tested with several distributed resource management system like Sun Grid Engine, Platform LSF, Torque/PBSPro, PBS Pro, Condor, Apple Xgrid and Simple Linux Utility for Resource Management (SLURM). The Advance Reservations capabilities were exposed for SGE, LSF and Maui (a scheduler that is typically used in conjunction with Torque systems).
- **QCG-Broker**
  The Grid Resource Management System (GRMS aka QCG-Broker [12]) is an open source meta-scheduling system, which allows developers to build and

deploy resource management systems for large scale distributed computing infrastructures. The QCG-Broker, based on dynamic resource selection, mapping and advanced scheduling methodology, combined with feedback control architecture, deals with dynamic Grid environment and resource management challenges. It is capable of load-balancing of jobs among clusters and co-allocating of resources. The main goal of the QCG-Broker is to manage the whole process of remote job submission to various batch queuing systems. It has been designed as an independent core component for resource management processes which can take advantage of various low-level core and Grid services responsible for the execution of jobs and reservation of resources on cluster machines. The QCG-Broker allows to co-allocate resources belonging to different e-infrastructures and execute cross-cluster and cross-infrastructure DRIHM applications.

4.2 Service Layer

The Service Layer has two peculiarities: the services on the Basic Service Layer, and the services on the Compound Service Layer. Since many services may exhibit functionalities to belong to either sublayer, we will not distinguish between both sublayers in this report unless stated otherwise.

The inventory on the Service Layer can be classified by services related to data management, services related to model management, and services related to workflow management.

4.2.1 Services related to data management

Data management in the sense of this report refers to event data acquisition, data format transformation, and data administration. Since these services typically rely on concrete data, generic services are not considered in this deliverable. Future versions may, however, include specific data management services – depending on the final decisions regarding the models to be considered in DRIHM.

4.2.2 Services related to model management

Model management in the sense of this report refers to the provisioning of models and their wrapping for interoperability purposes (typically done by scientists).

For interoperability purposes DRIHM will consider OpenMI [1]. The following description is adapted from [16]:

OpenMI focuses on the computation phase of applications. The code base that performs that computation is referred to as the *engine*. OpenMI also defines a *linkable component* as an engine that has been adapted to provide dynamic data exchange. Dynamic data is transferred between components by the linking of *output exchange items and input exchange items* (sometimes also referred to as *targets* and *sources*). A *model* is a specific instance of a component that also includes input data. For instance, a ground water model would be a ground water component with input data for a specific geographic region of interest. When two or more models are linked together they are considered to be in a *composition*.

This approach breaks the pattern of importing all input data before computation and not writing any outputs until computation has been completed. To reiterate, an OpenMI model component will only request certain inputs immediately prior to their use in the engine and provide some outputs immediately after they have been computed in the engine In order to achieve this, the application needs to be turned into an engine component that implements this dynamic data transfer in a way that other engines can also recognize and utilize.

The OpenMI addresses these issues by defining a set of software interfaces (the *OpenMI Standard*), which, if implemented around the engine, will allow other similarly modified engines to transfer data dynamically between each other. If an engine implements this standard in an approved way (i.e. as recognized by the OpenMI Association) the application is deemed to be *OpenMI compliant*. Users of a compliant model will know that they can perform dynamic data exchange with that engine. However, they will not necessarily know whether the data exchange is of a type or form that would be useful to their particular interests. Hence, an additional task of OpenMI is to define a uniform way of describing the available targets and sources. The standard is intended to enable on-line (memory based) data exchange between models on a time basis (e.g. a time stamp or a time span). Given that different models will be running on different temporal and spatial domains, the mechanism for data conversions between domains is also required. For example, spatial differences are handled in OpenMI 1.4 with Data Operations and in OpenMI 2.0 with the use of adaptors. It is the intention of the DRIHM project to use OpenMI 2.0. Consequently, the standard must not only specify definitions, but also provide guidelines as to how those definitions are implemented.

The standard is not limited to connecting time-based engines. Rather, version 2 of the standard has specifically concentrated on simplifying the mixing of temporal and non-temporal models (e.g databases) within a single composition.

Based on this description, OpenMI is of particular interest for DRIHM as DRIHM will have to cope with a set of heterogeneous models. For the initial version of this report the metadata pertaining to the initial list of candidate models is[5] as follows:

Meso-NH EPS

Identification info	
Title	Meso-NH EPS (DRIHM)
Date	
Date type	**Creation**: Date identifies when the resource was brought into existence
Abstract	A meteorological Ensemble Prediction System based on Meso-NH
Point of contact	

[5] The descriptions are copied from the FluidEarth catalogue (http://catalogue.fluidearth.net/) [5]

Organisation name	CNRS	**Electronic mail address**	
Role	**Author**: Party who authored the resource	**OnLine resource**	http://mesonh.aero.obs-mip.fr/mesonh/
Descriptive keywords			Meteorological Ensemble Prediction System .
Topic category code			Geoscientific information
Metadata			
File identifier			ac583761-721e-4f78-b5c1-445f63e19c80
Date stamp			2012-02-13T11:12:34
Contact			
Organisation name	CNRS	**Electronic mail address**	
Role	**Author**: Party who authored the resource		
Fluid Earth Model Engine - Technical Description			
Programming language(s) used to develop the model			
Programming language			Fortran
Platform(s) supported by the model			
Supported platform			BlueGene/P JUGENE, SGI Altix ICE, BULLX Nehalem, NEC SX8 & 9, etc.
Spatial dimension			4
Source code URI			http://mesonh.aero.obs-mip.fr/mesonh/
Executable URI			http://mesonh.aero.obs-mip.fr/mesonh/
Documentation			

Documentation URI	http://mesonh.aero.obs-mip.fr/mesonh/
Open MI status	Does not implement OpenMI
Open MI version	1.0.0
Number of processors	1-130000
Typical run time	
Time	several
unit	hour
Input	
Name	Initial and boundary conditions
Description	IFS, ARPEGE, ALADIN or AROME files
Format	GRIB
Mandatory	true
Input	
Name	Physiographic data
Description	Digital elevation model + land use
Format	HDR+DIR
Mandatory	true
Output	
Name	Meteorological fields
Description	include surface rainfall rate and accumulation, soil moisture, 2-m humidity, etc.
Format	FM
Mandatory	false

NMMB – Nonhydrostatic Multiscale Model on the B grid

Identification info	
Title	NMMB – Nonhydrostatic Multiscale Model on the B grid (DRIHM)
Date	18.08.2008
Date type	**Creation**: Date identifies when the resource was brought into existence

Abstract			NMMB is nonhydrostatic multiscale model on the B grid. Its unified non-hydrostatic dynamical core supports regional and global simulations and includes generalized hybrid vertical coordinate. Regional implementation of NMMB is current NCEP mesoscale model for operational weather forecasting in the United States of America. In high resolution Numerical Weather Prediction (NWP) applications, the computational efficiency of the model has been higher than the efficiency of most established non-hydrostatic models. The high computational efficiency of the NMM is primarily due to the design of the time-stepping procedure (Janjic, 2003). For more information check http://www.ecmwf.int/newsevents/meetings/workshops/2010/Non_hydrostatic_Modelling/presentations/Janjic.pdf
Point of contact			
Organisation name	Republic Hydromoeteorological Service of Serbia	**OnLine resource**	_
Role	**User**: Party who uses the resource		
Descriptive keywords			non-hydrostatic, multiscale, forecast .
Topic category code			Climatology, meteorology, atmosphere
Metadata			
File identifier			2cdaf60c-dd45-416e-80fd-69b3b2291cb3
Date stamp			2012-01-25T12:16:29
Contact			
Organisation name	Republic Hydromoeteorological Service of Serbia		

Role	**User**: Party who uses the resource		

Fluid Earth Model Engine - Technical Description	
Programming language(s) used to develop the model	
Programming language	Fortran 90
Platform(s) supported by the model	
Supported platform	Linux
Spatial dimension	3
Source code URI	not available yet
Executable URI	not available yet
Documentation	
Documentation URI	not available yet
Open MI status	Does not implement OpenMI
Open MI version	1.0.0
Number of processors	128
Typical run time	
Time	10800
unit	second
Input	
Name	pressure, u, v, temperature, relative humidity, geopotential height, cloud water, soil temperature, soil moisture, sea surface temperature, snow, sea ice
Description	The NMMB can be run using boundary condition created by its self (running as a global model and using any global analyses as an initial field) or using boundary condition from other global NWP model outputs (e.g. ECMWF or GFS). For any of these cases preferred input file format is GRIB1 or GRIB2 but preprocessor can be easily redesigned to use other input file formats, e.g. NetCDF. Several nesting stages are possible using boundary condition from parent integration.
Format	netCDF, Grib1 or Grib2
Mandatory	false
Output	
Name	precipitation, wind, temperature, heat and moisture fluxes, water vapour

Description	model output
Format	netCDF, GRIB1
Mandatory	false

PhaSt

<table>
<tr><td colspan="5">Identification info</td></tr>
<tr><td colspan="3">Title</td><td colspan="2">PhaSt (DRIHM)</td></tr>
<tr><td colspan="3">Date</td><td colspan="2">09.12.2011</td></tr>
<tr><td colspan="3">Date type</td><td colspan="2">Creation: Date identifies when the resource was brought into existence</td></tr>
<tr><td colspan="3" rowspan="3">Abstract</td><td colspan="2">the PhaSt (Phase Stochastic) model is a rainfall nowcasting method based on the combination of an empirical nonlinear transformation of measured precipitation fields and the stochastic evolution in spectral space of the transformed fields is</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2">introduced. The power spectrum and the amplitude distribution of precipitation are kept constant during the forecast, and a Langevin-type model is used to evolve the Fourier phases.</td></tr>
<tr><td colspan="5">Point of contact</td></tr>
<tr><td>Organisation name</td><td>CIMA Research Foundation</td><td>Electronic mail address</td><td colspan="2"></td></tr>
<tr><td>Role</td><td>Author: Party who authored the resource</td><td>OnLine resource</td><td colspan="2">www.cimafoundation.org</td></tr>
<tr><td colspan="3">Descriptive keywords</td><td colspan="2">Stochastic Ensemble Nowcasting Model .</td></tr>
<tr><td colspan="3">Topic category code</td><td colspan="2">Geoscientific information</td></tr>
<tr><td colspan="5">Metadata</td></tr>
<tr><td colspan="3">File identifier</td><td colspan="2">e3a545ca-6a3d-40b8-bd61-22955f8c1f05</td></tr>
</table>

Date stamp	2011-12-12T15:40:54		
Contact			
Organisation name	CIMA Research Foundation	**Electronic mail address**	
Role	**Author**: Party who authored the resource		
Fluid Earth Model Engine - Technical Description			
Programming language(s) used to develop the model			
Programming language	Matlab		
Platform(s) supported by the model			
Supported platform	Linux		
Platform(s) supported by the model			
Supported platform	Windows		
Spatial dimension	2		
Open MI status	Does not implement OpenMI		
Open MI version	1.0.0		
Typical run time			
Time	180		
unit	second		
Input			
Name	Radar Rainfall Maps		
Description	last two rainfall maps observed by a meteorological radar		
Format	Binary/Ascii raster		
Mandatory	true		
Output			
Name	Ensemble Precipitation Nowcasting Fields		
Description	Ensemble of possible evolutions of the precipitation field in the next 1-2hours		
Format	Binary/Ascii raster		
Mandatory	true		

DRIHM
DISTRIBUTED RESEARCH INFRASTRUCTURE FOR HYDRO-METEOROLOGY

RainFARM

<table>
<tr><td colspan="4">Identification info</td></tr>
<tr><td colspan="3">Title</td><td>RainFARM (DRIHM)</td></tr>
<tr><td colspan="3">Date</td><td>09.12.2011</td></tr>
<tr><td colspan="3">Date type</td><td>Creation: Date identifies when the resource was brought into existence</td></tr>
<tr><td colspan="3">Abstract</td><td>The Rainfall Filtered Autoregressive Model (RainFARM) is a method for stochastic rainfall downscaling that can be easily applied to the precipitation forecasts provided by meteorological models. RainFARM is based on the nonlinear transformation of a Gaussian random field, and it conserves the information present in the rainfall fields at larger scales.</td></tr>
<tr><td colspan="4">Point of contact</td></tr>
<tr><td>Organisation name</td><td>CIMA Research Foundation</td><td>Electronic mail address</td><td></td></tr>
<tr><td>Role</td><td>Author: Party who authored the resource</td><td>OnLine resource</td><td>www.cimafoundation.org</td></tr>
<tr><td colspan="3">Descriptive keywords</td><td>Stochastic Rainfall Downscaling Model .</td></tr>
<tr><td colspan="3">Topic category code</td><td>Geoscientific information</td></tr>
<tr><td colspan="4">Metadata</td></tr>
<tr><td colspan="3">File identifier</td><td>7afbf0f9-07bf-49c4-94b3-26b0103c8c84</td></tr>
<tr><td colspan="3">Date stamp</td><td>2011-12-12T15:38:25</td></tr>
<tr><td colspan="4">Contact</td></tr>
</table>

Role	**Author**: Party who authored the resource	**Electronic mail address**	
Fluid Earth Model Engine - Technical Description			
Programming language(s) used to develop the model			
Programming language		Matlab	
Programming language(s) used to develop the model			
Programming language		C++	
Platform(s) supported by the model			
Supported platform		Linux	
Platform(s) supported by the model			
Supported platform		Windows	
Spatial dimension		2	
Open MI status		Does not implement OpenMI	
Open MI version		1.0.0	
Typical run time			
Time		180	
unit		second	
Input			
Name		Low resoution precipitation field [P(x,y,t)]	
Description		Low resolution (10-15 km/3-12h) precipitation field, tipicallyLimited area model or GCM output	
Format		Binary/Ascii raster	
Mandatory		true	
Output			
Name		High resolution ensemble	
Description		Ensemble of (tipically 100) fine resoution (1km, 10 minutes) precipitation fields	
Format		Binary/Ascii raster	
Mandatory		true	

DRiFt

DRIHM
DISTRIBUTED RESEARCH INFRASTRUCTURE FOR HYDRO-METEOROLOGY

<table>
<tr><td colspan="4">Identification info</td></tr>
<tr><td colspan="3">Title</td><td>DRiFt (DRIHM)</td></tr>
<tr><td colspan="3">Date</td><td>12.12.2011</td></tr>
<tr><td colspan="3">Date type</td><td>Creation: Date identifies when the resource was brought into existence</td></tr>
<tr><td colspan="3" rowspan="3">Abstract</td><td>DRiFt (Discharge River Forecast) is a linear, semi-distributed event model based on a geomorphologic approach. The model is focused on the efficient description of the drainage system in its essential parts: hillslopes and channel networks. These are addressed with two kinematic scales, which determine the base of the geomorphologic response of the basin.</td></tr>
<tr><td></td></tr>
<tr><td>The morphologic module is coupled with a simple representation of soil infiltration properties (SCS-CN, 1985). Discharge at any location along the drainage network is evaluated by applying the convultion integral of the time-variant Instantaneous Unit Hydrograph</td></tr>
<tr><td colspan="4">Point of contact</td></tr>
<tr><td>Organisation name</td><td>CIMA Foundation</td><td>Electronic mail address</td><td></td></tr>
<tr><td>Role</td><td>Author: Party who authored the resource</td><td>OnLine resource</td><td>www.cimafoundation.org</td></tr>
<tr><td colspan="3">Descriptive keywords</td><td>flood forecasting .</td></tr>
<tr><td colspan="3">Language</td><td>English</td></tr>
<tr><td colspan="3">Topic category code</td><td>Geoscientific information</td></tr>
<tr><td colspan="4">Metadata</td></tr>
<tr><td colspan="3">File identifier</td><td>9c787ae2-7bbe-4de1-bb98-52fd32279fed</td></tr>
<tr><td colspan="3">Date stamp</td><td>2011-12-12T14:30:56</td></tr>
<tr><td colspan="4">Contact</td></tr>
</table>

DRIHM

DISTRIBUTED RESEARCH INFRASTRUCTURE FOR HYDRO-METEOROLOGY

Organisation name	CIMA Foundation	**Electronic mail address**	
Role	**Author**: Party who authored the resource		
Fluid Earth Model Engine - Technical Description			
Programming language(s) used to develop the model			
Programming language	Fortran		
Platform(s) supported by the model			
Supported platform	Windows		
Platform(s) supported by the model			
Supported platform	Linux		
Spatial dimension	2		
Source code URI	-		
Executable URI	-		
Documentation			
Documentation URI	-		
Open MI status	Does not implement OpenMI		
Open MI version	1.0.0		
Number of processors	-		
Typical run time			
Time	-		
unit	second		
Input			
Name	DTM		
Description	Grid DTM		
Format	Any Raster Format		
Mandatory	true		
Input			
Name	Land Cover/Use		
Description	Land Cover/Use description in terms of SCS-CN		
Format	Any raster Format		
Mandatory	true		

Input	
Name	Rainfall
Description	Rainfall map
Format	ascii or binary raster
Mandatory	true
Output	
Name	Q
Description	Discharge
Format	ascii time series
Mandatory	false

Inundator

Identification info			
Title		Inundator (DRIHM)	
Date		12.12.2011	
Date type		**Creation**: Date identifies when the resource was brought into existence	
Abstract		Inundator is a 2D inundation model based on an approximation of the 2D shallow water equations. It uses a simplified set of Initial and Boundary conditions and solves some numerical instability problems by applying physically based flux limiter equations (e.g., the spillway equation).	
Point of contact			
Organisation name	CIMA Foundation	**Electronic mail address**	
Role	**Author**: Party who authored the resource	**OnLine resource**	www.cimafoundation.org
Descriptive keywords		flood, flood water depth, shallow water equations .	
Language		English	
Topic category code		Geoscientific information	

Metadata			
File identifier	49a15cd7-6a74-4676-ba19-860339563f12		
Date stamp	2011-12-12T14:04:36		
Contact			
Organisation name	CIMA Foundation	**Electronic mail address**	
Role	**Author**: Party who authored the resource		
Fluid Earth Model Engine - Technical Description			
Programming language(s) used to develop the model			
Programming language	Matlab script		
Platform(s) supported by the model			
Supported platform	Windows		
Platform(s) supported by the model			
Supported platform	Linux		
Spatial dimension	2		
Source code URI	-		
Executable URI	-		
Documentation			
Documentation URI	-		
Open MI status	Does not implement OpenMI		
Open MI version	1.0.0		
Number of processors	-		
Typical run time			
Time	-		
unit	second		
Input			
Name	DTM		
Description	Grid DTM		
Format	Any Raster Format		

Mandatory	true
Input	
Name	Land Cover/Use
Description	Land Cover/Use description in terms of hydraulic roughness / anderson category
Format	Any raster Format
Mandatory	true
Input	
Name	Flood Hydrograph
Description	Flood Hydrograph time series
Format	ascii time series
Mandatory	true
Output	
Name	Water Depths
Description	Water Depths
Format	Any Raster Format
Mandatory	false
Output	
Name	Water Velocity
Description	Water Velocity
Format	Any Raster Format
Mandatory	false

Contiuum

Identification info	
Title	Contiuum (DRIHM)
Date	12.12.2011
Date type	**Creation**: Date identifies when the resource was brought into existence

Abstract			Continuum is a continuous distributed hydrological model that strongly relies on a morphological approach. All the main hydrological phenomena are modeled in a distributed way. Mass balance and energy balance are solved explicitly so that state variables such as land surface temperature, which is particularly suitable to be extensively observed and assimilated, are computed. The model is based on a small number of territorial data. This allows for its implementation in a wide range of environments with reduced effort. It reproduces the most important hydrological processes: vegetation retention and evaporation, infiltration and runoff generation, channel and overland flow, subsurface and deep flow separation, water table recharge and evapotranspiration.
Point of contact			
Organisation name	CIMA Foundation	**Electronic mail address**	
Role	**Author**: Party who authored the resource	**OnLine resource**	www.cimafoundation.org
Descriptive keywords			flood, distributed modeling, land surface temperature, energy balanceperature .
Topic category code			Geoscientific information
Metadata			
File identifier			66a24261-53b4-4927-8f2f-33a7ccfab283
Date stamp			2011-12-12T12:17:42
Contact			
Organisation name	CIMA Foundation	**Electronic mail address**	
Role	**Author**: Party who authored the resource		

Fluid Earth Model Engine - Technical Description	
Programming language(s) used to develop the model	
Programming language	Fortran
Platform(s) supported by the model	
Supported platform	Linux
Platform(s) supported by the model	
Supported platform	Windows
Spatial dimension	2
Source code URI	-
Executable URI	-
Documentation	
Documentation URI	-
Open MI status	Does not implement OpenMI
Open MI version	1.0.0
Number of processors	-
Typical run time	
Time	15 sec. for a day of simulation
unit	second
Input	
Name	Rainfall, Temperature, Relative Humidity, Wind speed, Shortwave Solar Radiation
Description	Meteorological input
Format	Binary raster
Mandatory	false
Input	
Name	DTM, Soil Use, Soil Type
Description	Land data
Format	ASCII or Binary raster
Mandatory	false
Output	
Name	Q_BasinName
Description	Discharge
Format	ASCII time series
Mandatory	false
Output	

Name	LST_BasinName
Description	Land Surface Temperature
Format	ASCII or Binary raster
Mandatory	false
Output	
Name	SM_BasinName
Description	Soil Moisture as soil saturation
Format	ASCII or Binary raster
Mandatory	false

4.2.3 Services related to workflow management

For the initial version of this report, workflow management consists of combining models and basic services. The following tools are important for DRIHM:

- **Pipistrelle**

  Pipistrelle (http://openweb.uk.net/Training/Welcome.aspx) is a GUI for composing and running OpenMI model compositions.

- **SPRUCE**

  The traditional high performance computing batch queue model is not adequate enough for DRIHM simulations to be prioritized by their urgency. Typically a Grid will provide general purpose resources to a wide range of different users. If these resources are to be used by legal authorities, fast reactions in support of sudden events, especially in support of emergency intervention planning, then some way is needed of prioritizing simulations above the normal workload on a computational resource. SPRUCE, A System for Supporting Urgent High-Performance Computing [11], developed at Argonne National Labs, USA, is a tool which allows this to happen. Uers with simulations that are considered an emergency are issued with SPRUCE tokens, which allow them to submit emergency jobs to a dedicated machine. The SPRUCE middleware takes care of running the job in a high priority mode, pre-empting the work that is already running on the machine.

- **GridSpace**

  GridSpace [11] is a novel virtual laboratory framework enabling researchers to conduct virtual experiments including running of HMR applications on Grid-based resources and other HPC infrastructures. The current generation of GridSpace - GridSpace2 - facilitates the exploratory development of experiments by means of scripts which can be expressed in a number of popular languages, including Ruby, Python and Perl. In addition, GridSpace2 provides a Web 2.0-based Experiment Workbench supporting joint development and execution of virtual experiments by groups of collaborating scientists.

4.3 Application Layer

On the Application Layer DRIHM requires to run applications belonging to the experiment suites [1] dealing with heterogeneous models on resources provided by national and international Grids, in addition to local departmental and institutional clusters, while hiding from the user the details of the underlying middleware in use by the Grid. The University College London has developed with the Application Hosting Environment (AHE) [8] a simple desktop and command line interface, to run DRIHM applications on resources provided by heterogeneous providers, while hiding from the user the details of the underlying Grid middleware. In addition, a mobile interface for Windows Mobile based PDAs is available, and an iPhone interface is in development. The AHE is able to run applications on both Unicore-based and Globus Toolkit-based Grids in an interoperable way, meaning that a user can use a single AHE installation to access resources from Globus-based NGIs and DEISA/PRACE. The development of an EGI connector for AHE is currently underway.

AHE is designed to allow scientists to quickly and easily run unmodified, legacy applications on Grid resources, manage the transfer of files to and from the Grid resource and monitor the status of the scheduled application. The philosophy of the AHE is based on the fact that very often a group of researchers will all want to access the same application, but not all of them will possess the skill or inclination to install the application on remote grid resources. In the AHE, an expert user installs the application and configures the AHE server, so that all participating users can share the same application.

4.4 Access Layer

On the Access Layer DRIHM requires to provide hydro-meteo scientists with an easy and intuitive access to the resources, i.e., data, models and computing infrastructures, in a seamless way irrespective of their location. We identified science gateway technologies as the most promising solution for this issue. A Science Gateway is a portal which integrates the toolsets, the data and whatever else is necessary for a scientific community to carry out their research using Grid or Cloud infrastructures. As a result Science Gateways will enable scientists to use resources in an easy and intuitive way, allowing them to focus on their research aspects instead on the learning and managing complex infrastructures. While in some e-science infrastructures, e.g., the Extreme Science and Engineering Discovery Environment (XSEDE), the gateway is a mature and well accepted technology (see https://www.xsede.org/gateways-overview) , in Europe EGI is promoting a set of workshops aiming at supporting the development of tools for EGI Virtual Research Communities through sharing best practices[6].

---

[6] https://www.egi.eu/indico/sessionDisplay.py?sessionId=64&confId=452#20110920

There are two basic options to build a Science Gateway: either to build it from scratch or to adopt and customize an existing gateway technology. Moreover, gateways can be categorized in three classes according to the way the user connects to the portal and to which resources the portal can connect to in the back end:

- Web portal: the user interface is a Web browser-based application with users in front and Grid services in back.
- Desktop application: the interface is an application or suite of applications that run directly on the users' machines and that access Grid services.
- Grid-bridging gateway: some communities run their own Grids that are devoted to their areas of science. In these cases, the gateway is a mechanism to extend the reach of their community Grid so its users can also use the resources of a larger Grid.

Disregarding the solution to build a gateway from scratch, DRIHM is presently evaluating the most interesting Gateways toolkits in order to select the one that best fits the DRIHM requirements and the set of services that will be deployed over the DRIHM architecture.

5 Conclusion

This document describes the inventory of services necessary for successful HMR over Grids. The inventory listed here refers to an initial setup of the DRIHM reference stack as depicted in Figure 3.

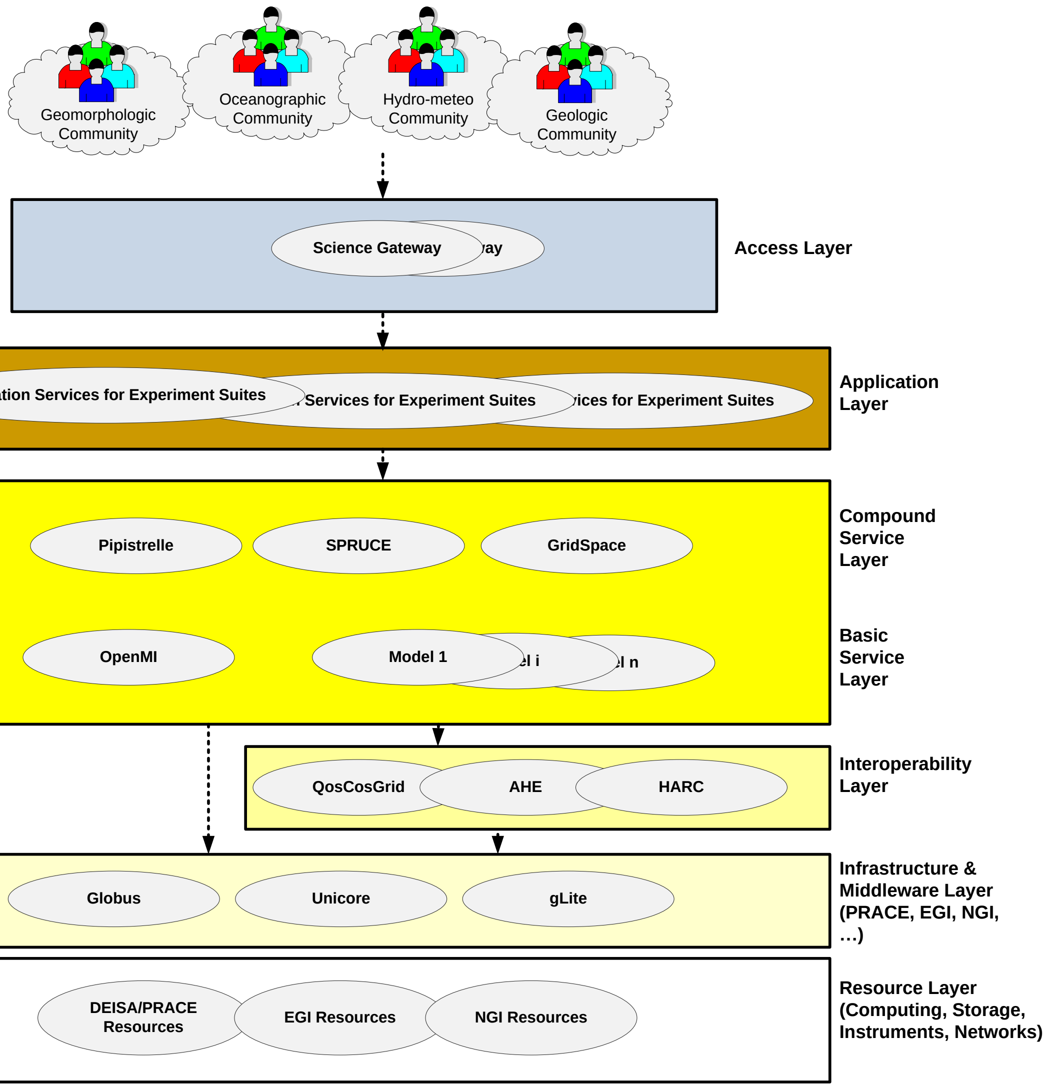

Figure 3: DRIHM service inventory

Changes to this inventory (i.e., addenda or removals) will be reflected in future issues of this deliverable. The main objective, however, is to keep the inventory small in order to avoid the risk of services not working together well and thus generating higher ongoing support costs.

One of the main challenges for the DRIHM infrastructure will be in handling the different technical and functional natures of the HMR models to be incorporated.

6 Acronyms and References

6.1 Acronyms

DEISA	Distributed European Infrastructure for Supercomputing Applications
EGI	European Grid Initiative
ESFRI	European Strategy Forum on Research Infrastructures
HMR	Hydro-Meteorologic Research
HPC	High Performance Computing
NGI	National Grid Initiative
PRACE	Partnership for Advanced Computing in Europe
VO	Virtual Organisation

6.2 References

[1] DRIHM Description of Work, Part B, 2011

[2] A. Parodi et al.: The DRIHMS White Paper, http://www.drihms.eu/publications/material/White%20Papers%20Drihms_final.pdf, 2010

[3] MAPPER Deliverable D5.1: Report on the Inventory of Deployed Services, (http://www.mapper-project.eu/documents/10155/23586/D5.1+-+Deployed+Services.pdf, 2011

[4] DRIHM Deliverable D5.1: Report on the Assessment of Operational Procedures and Definition of the DRIHM Operational Model, 2012

[5] The Fluidearth Database: http://catalogue.fluidearth.net/)

[6] The HyMeX Database: http://mistrals.sedoo.fr/HyMeX

[7] OpenMI: http://www.openmi.org/reloaded/

[8] Zasada, S. J., & Coveney, P. V. (2009). Virtualizing access to scientific applications with the Application Hosting Environment (Vol. 180). Computer Physics Communications.

[9] MacLaren, J. (2007). HARC: The Highly-Available Resource Co-allocator. Proceedings of GADA'07 (OTM Conferences 2007, Part II) (pp. 1385-1402). Springer-Verlag.

[10] J. Gray and L. Lamport. Consensus on transaction commit. ACM TODS, 31(1):130–160, March 2006.

[11] Ciepiela, E., Harezlak, D., Kocot, J., Bartynski, T., Kasztelnik, M., Nowakowski, P., et al. Programming in the Virtual Laboratory. Proceedings of the International Multiconference on Computer Science and Information Technology, (pp. 621-628).

[12] Kurowski, K., Piontek, T., Kopta, P., Mamoński, M., & Bosak, B. (2010). Parallel Large Scale Simulations in the PL-Grid Environment. Computational Methods in Science and Technology , 47-56.

[13] Caromel, D., Delbe, C., di Costanzo, A. and Leyton, M.: ProActive: an Integrated Platform for Programming and Running Applications on Grids and P2P systems. Computational Methods in Science and Technology, vol. 12, no. 1, pp. 69-77 (2006).

[14] GFD 114: HPC Basic Profile, Version 1.0, http://www.ogf.org/documents/GFD.114.pdf

[15] GFD 56: Job Submission Description Language (JSDL) Specification, Version 1.0 - http://www.gridforum.org/documents/GFD.56.pdf.

[16] The OpenMI Association Technical Committee: OpenMI Document Series: The OpenMI 'in a Nutshell' for the OpenMI (Version 2.0), http://www.openmi.org/reloaded/about/publications-documents/v20/OpenMI+In+a+Nutshell.pdf, 2010