DRIHM2US

DISTRIBUTED RESEARCH INFRASTRUCTURE FOR HYDRO-METEOROLOGY TO UNITED STATES OF AMERICA

D2.2: Report on a Common Architecture Model

Abstract: This document proposes a common architecture model for HMR infrastructures. It is based on an analysis conducted in a previous deliverable.

DRIHM2US (G.A. n° 313122) is co-Funded by the EC under 7th Framework Programme

Document Information Page

Contract Number	313122
Project Name	Distributed Research Infrastructure for Hydro-Meteorology to United States of America
Project Acronym	DRIHM2US
Deliverable Number	2.2
Deliverable Name	Report on a Common Architecture Model
Work Package Number	2
Work Package Name	Architecture Harmonization Analysis and Planning
Deadline	01/10/2013
Version	1.0
Dissemination Level	PU
Nature	R
Lead Beneficiary	LMU

DRIHM2US

Document History

Date	Version	Description
4th Sept 2013	0.8	Initial write up for review by the project team.
13th Oct 2013	0.9	Handed over to reviewers
10th Nov 2013	1.0	Integrated reviewer's comments

Table of Contents

List of Figures

1 Executive Summary

The objectives of work package 2 (Architecture Harmonization Analysis and Planning) are to analyze different e-infrastructure approaches in Europe and in the US as far as they relate to hydro-meteorological research, and to propose a common architecture model for HMR infrastructures. This report should be considered as addendum to report D2.1 in which various e-infrastructure approaches were analyzed [3]. Based on the reference framework defined in [3] this report specifies an appropriate common architecture model using the Model Driven Architecture (MDA) approach.

In the report we outline an HMR domain model, a use case model and a static class model of what we call an "HMR Science System". Both the dynamic interaction models and the physical deployment model are briefly touched upon but are omitted in this report as they need further decision making. These models will therefore be provided later as an addendum to this report.

The findings of task 2.2 will now be fed into a gap analysis (task 2.3).

2 Introduction

The emergence of data driven science reflects the increasing value of observational, sensor, streaming and experimental data in every field of science. Hence, information and communication technology infrastructures supporting scientific data handling and workflow processes are gaining more and more importance. However, often data can neither be shared nor are data consuming applications and models interoperable across countries and disciplines; moreover, both are mostly unsustainable due to the lack of commonly agreed governance policies, inadequate legal frameworks and short-sighted funding models [7].

It is envisioned that by 2020/2030 all stakeholders, from scientists, science managers, infrastructure operators and governmental authorities to the general public, are aware of the critical importance of preserving and sharing reliable data collected by a vast array of sensors and instruments during scientific processes and everyday life behavior [9] [14]. Unfortunately, however, diversity is likely to remain a dominant feature of scientific information: not only diversity of formats and types but also of people and communities that generate and use scientific data.

It is therefore of paramount interest that user communities in the various fields of science and humanities, research facilities producing data, and international teams of scientists should take a more active role in the definition of concrete short and long term requirements for the underlying scientific data infrastructures. Federated repositories, libraries and data centers should be interoperable at a global level with high degrees of dependability and trust, guided by international standards. Additionally, because all research builds on earlier work, not only fuller and wider access to scientific data is a prerequisite for accelerating scientific progress, but also a well concerted interplay between all stakeholders as exemplified in Figure 1 for the European Grid Infrastructure (EGI) ecosystem [10].

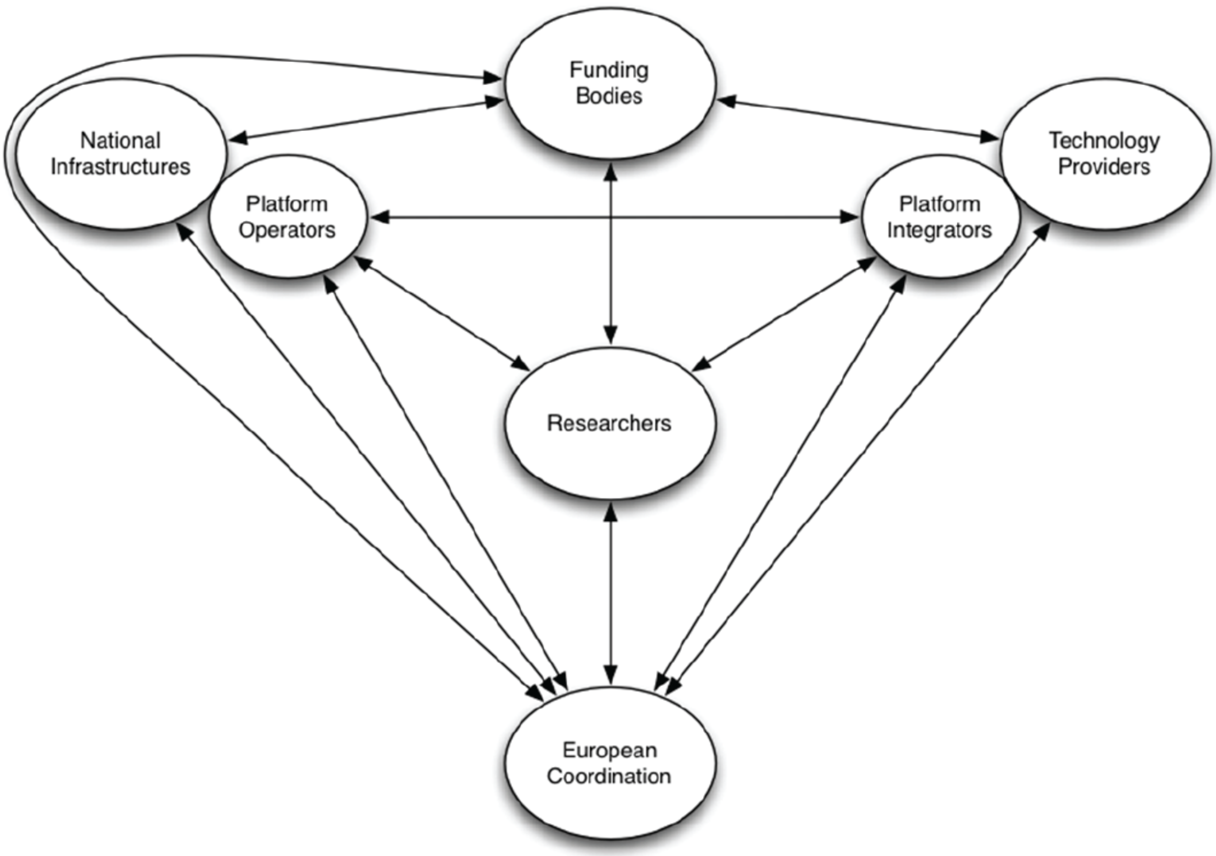

Figure 1: EGI Ecosystem [10]

The overarching goal of the DRIHM2US project is to support this vision. DRIHM2US aims at promoting the international cooperation between Europe and the USA for the development of a global, interoperable, sustainable common e-infrastructure for Hydro-Meteorological Research (HMR) by facilitating the persistent availability and an effective sharing of data and models across scientific disciplines, institutions and national boundaries [1].

HMR is an area of critical scientific importance and of high societal relevance. It plays a key role in guiding predictions relevant to the safety and prosperity of humans and ecosystems from highly urbanized areas to coastal zones and agricultural landscapes. Although it is well understood that hydro-meteorological events have to be investigated along the geographic

scale (local, regional, global) [2], it is on the other hand necessary to better understand the commonalities and differences of e-infrastructures supporting this endeavor (which was the focus of report [3]) and to eventually define a "best practice" HMR e-infrastructure implied by a common architecture model. The specification of this model is the primary objective of this report.

In order to better understand the ratio behind the proposed model we refer the reader to report [3].

The report is organized as follows: In section 3 we discuss the terms we will be using and the model driven methodology of specifying an architecture. Section 4 applies this methodology to define a platform independent model of the DRIHM2US common architecture by elaborating a domain model, a use case model, logical class model. We also touch informally a dynamic model and a physical deployment model. Section 5 concludes the report.

3 Modelling Background

3.1 General Terms

Research in general and HMR in particular is increasingly based on distributed regional, national and global collaborations of scientists facilitated by the Internet as a global communication infrastructure and by cooperation mechanisms following the Grid Computing paradigm [4], the Cloud Computing paradigm [5] or a mix of both.

"*e-Infrastructure*" is the term used for the technology and the organizational model that support research undertaken in this way [6]. Consequently, it embraces networks, Grids, data centers, collaborative environments, service registries, single-sign on mechanisms, certificate authorities, training and help-desk services. Most importantly, however, it is the seamless integration of these concepts that defines an e-infrastructure. Community-specific e-infrastructures (like the ones relevant for HMR) also feature community-specific mechanisms like model coupling or data format conversions.

Based on this notion of e-infrastructure, an *architecture model* in the sense of this report renders more precisely the organizational structure and the associated behavior of IT systems. The architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling the parts. In contrast to this and on a more abstract level, a *framework* contains model elements that specify a reusable architecture for all or part of a system. When frameworks are specialized for an application domain, they are sometimes referred to as application frameworks – or in the special case of this report HMR application framework or simply HMR framework.

Since we are interested in modelling a common HMR architecture, the above terms are to be understood from a pure modelling perspective. Consequently, the reference framework defined in [3] and summarized in the following section is -- strictly speaking -- neither a framework nor is it an architecture. Instead, it serves as a classification scheme for HMR services, be they

generic ones or particular ones.

3.2 Model Driven Architecture (MDA)

In this report we use object oriented modelling with Unified Modelling Language (UML) constructs [18]. The modelling process itself follows the Model Driven Architecture (MDA) methodology [17]. For more details we refer the reader to the respective literature.

MDA developed by the Object Management Group (OMG)[1], is a framework for software development using a system modeling language and separating system architectures from platform architectures. The primary components of MDA technologies are the Platform Independent Models (PIM) and the Platform Specific Models (PSM). PIMs describe the structure and function of a system, but not the specific implementation.

MDA has the capability to define templates that map transformations from PIMs to PSMs. This facilitates the development of a system in abstraction, and simplifies implementation of that system across a variety of platforms. For instance, an MDA transformation from PIM to a DDL (Data Definition Language) will create DDL table elements from a class, whereas the same class transformed to an EJB (Enterprise JavaBeans) Entity Bean will result in a package containing the class and interface elements required by EJB. Figure 2 summarizes the relationships.

According to OMG, UML is a "graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The UML offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components."

[1] http://www.omg.org/

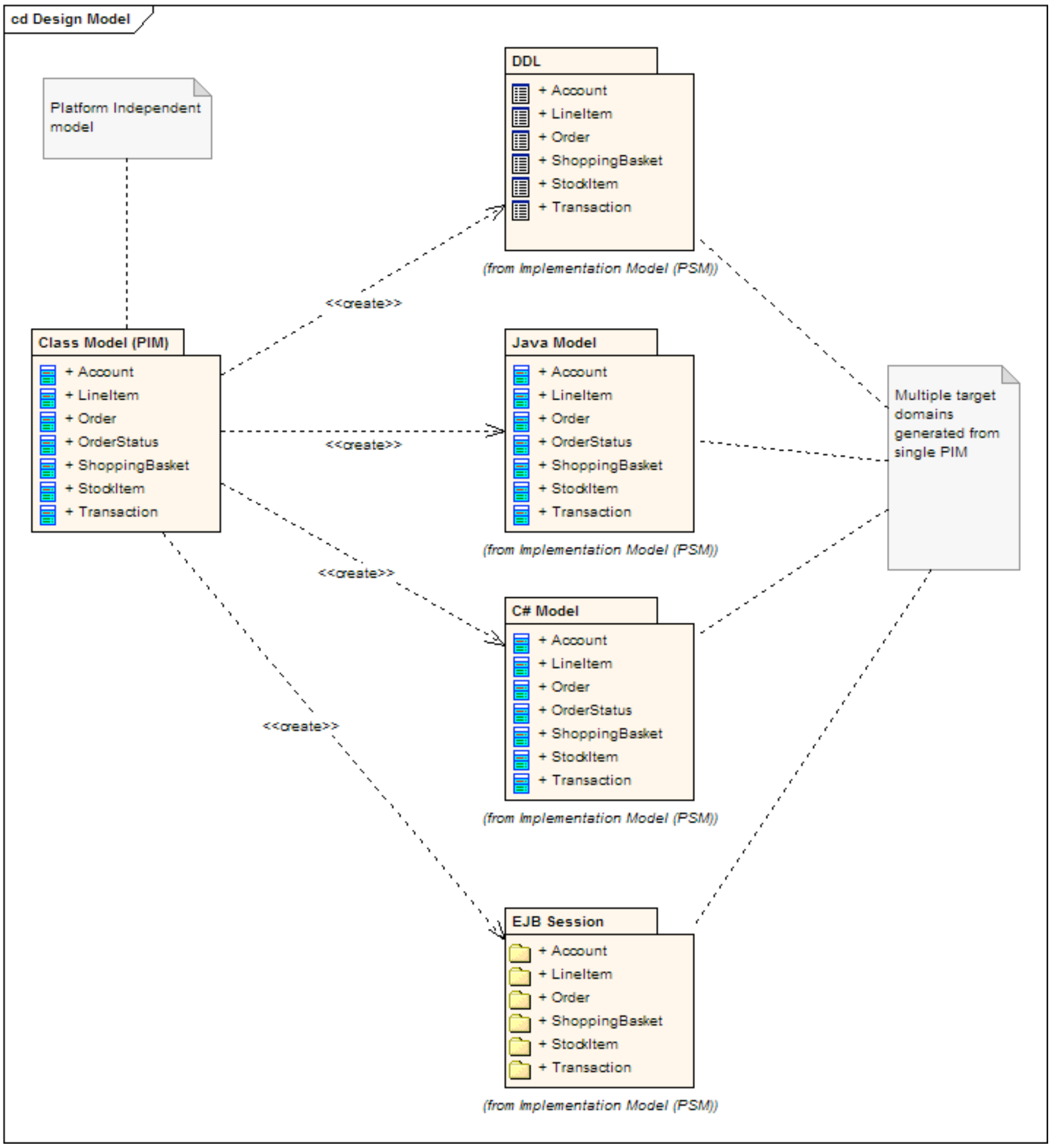

Figure 2: Example PIM, PSM, Transforms in MDA[2]

[2] Source: http://www.sparxsystems.com.au/resources/mda/

The important point to note here is that UML is a language for specifying and not a method or a procedure. The UML may be used in a variety of ways to support a software development methodology but in itself it does not specify that methodology or process.

UML defines the notation and semantics for the following domains:

- The User Interaction or Use Case Model describes the boundary and interaction between the system and users. It thus corresponds in some respects to a requirements model.
- The Interaction or Communication Model describes how objects in the system will interact with each other to get work done.
- The State or Dynamic Model describes the states or conditions that classes assume over time. Activity graphs describe the workflows the system will implement.
- The Logical or Class Model describes the classes and objects that will make up the system.
- The Physical Component Model describes the software (and sometimes hardware components) that make up the system.
- The Physical Deployment Model describes the physical architecture and the deployment of components on that hardware architecture.

The UML also defines extension mechanisms to meet specialized needs (for example Science Process modelling extensions or HMR specific extensions).

While MDA and UML typically consider business processes, for the convenience of this report we extend this notion to science processes. A Science Process Model deals with science processes which themselves

- Have a goal
- Have specific inputs
- Have specific outputs
- Use resources
- Have a number of activities that are performed in some order
- May affect more than one organizational unit, potentially geographically dispersed

• Create value of some kind for the scientists who may be internal or external.

Figure 3 reflects this process model in the context of business-oriented enterprise architectures.

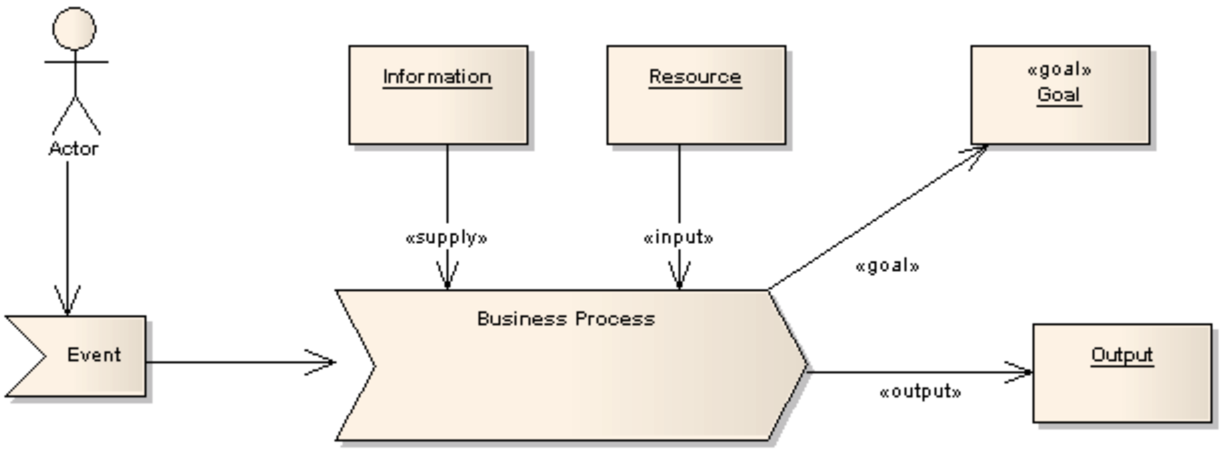

Figure 3: Abstract process model[3]

3.3 Methodology

The objective of this report is not to provide a complete architectural model for e-science (i.e., science based e-infrastructures). However, it is our intention to pave the way for a complete development cycle for a generic HMR e-infrastructure. Therefore, the following sections contain

- A *Use Case Model* to describe the proposed functionality of the HMR e-infrastructure
- A draft of a *Requirements Model*.

[3] Source: http://www.sparxsystems.com.au/downloads/whitepapers/The_Business_Process_Model.pdf

- An *HMR domain model*.
- A *logical model* as a static view of the objects and classes that make up the design/analysis space.
- A *dynamic model* to express and model the behavior of the system over time.
- A draft of a *Physical/Deployment Model* to provide a detailed view of the way components will be deployed across e-infrastructures.

The goal is that the entirety of these models (and further specifications in deliverables D2.3 and D2.4) makes up the PIM part of MDA. PSM specifications and PIM/PSM transformers will not be covered in this report.

4 Proposal of a Common Architecture Model

4.1 HMR e-Infrastructure Reference Framework Reloaded

In this section we briefly repeat [3] and set the context for the subsequent MDA/UML approach.

In their broadest possible terms, e-infrastructures deal with different stakeholders, data types and services which somehow interrelate to facilitate global science processes. Data generators (like sensors or instruments) and users (like scientists or citizen scientists [14]) gather, capture, transfer and process data – often across the globe in virtual research communities. They draw upon support services in their specific scientific communities (e.g., HMR) which typically comprise tools to locate data, process it, annotate it or interpret it. The support services themselves make use of a broad set of common data services including data storage, data identification, data authentication, data mining, and workflow/task execution. Based on these observations we propose the generic classification scheme of Figure 4 which also reflects the findings and suggestions in

- The European Commission High Level Expert Group on Scientific Data report "Riding the Wave" from October 2010 [13]
- The European Commission report "Open Infrastructures for Open Science – Horizon 2020 consultation report" by Richard L. Hudson and Carlos Morais Pires [9]
- The 2009 White Paper "Strategy for a European Data Infrastructure" of the European data initiative PARADE (the Partnership for Accessing Data in Europe) [15]
- Wilkins-Diehr, Nancy, Dennis Gannon, Gerhard Klimeck, Scott Oster and Sudhakar Pamidighantam: TeraGrid science gateways and their impact on science, published in Computer, 41(11):32{41, 2008} [11]
- The XSEDE architecture ratio as described in [16]
- The Second European Union-Australia Workshop on Research Infrastructure [8]

The scheme follows a simple 3-tier approach:

1. The User Tier comprises all building blocks relevant to the user working with the HMR e-infrastructure. Note that users may be scientists, administrators, citizen scientists, software processes, and sensors. Accordingly, access may be facilitated by command line interfaces, portals, GUIs, APIs, or alike.
2. Applications somehow use models and data. This usage pattern is reflected in the HMR Community Tier building blocks where applications draw upon HMR specific support services. The support services themselves make use of common (i.e., generic) data/model services that include functionalities to store and identify data (including archiving and Persistent Identifiers[4]), authenticate it, execute tasks, and mine it. Common services also include data conversion tools, model coupling mechanisms and generic visualization methods.
3. The Infrastructure Tier provides the physical resources and the mechanisms to access and manage these resources. The resources may in total or in parts be "contributed to" the National Grid Infrastructures (NGI), to the European Grid Infrastructure (EGI), to the Partnership for Advanced Computing in Europe (PRACE), to the Extreme Science and Engineering Discovery Environment (XSEDE) or others. Access is typically enabled by middleware technology assisted by metascheduler technology and application hosting environments – and preferably all based on standards.

---

[4] e.g., http://www.handle.net/

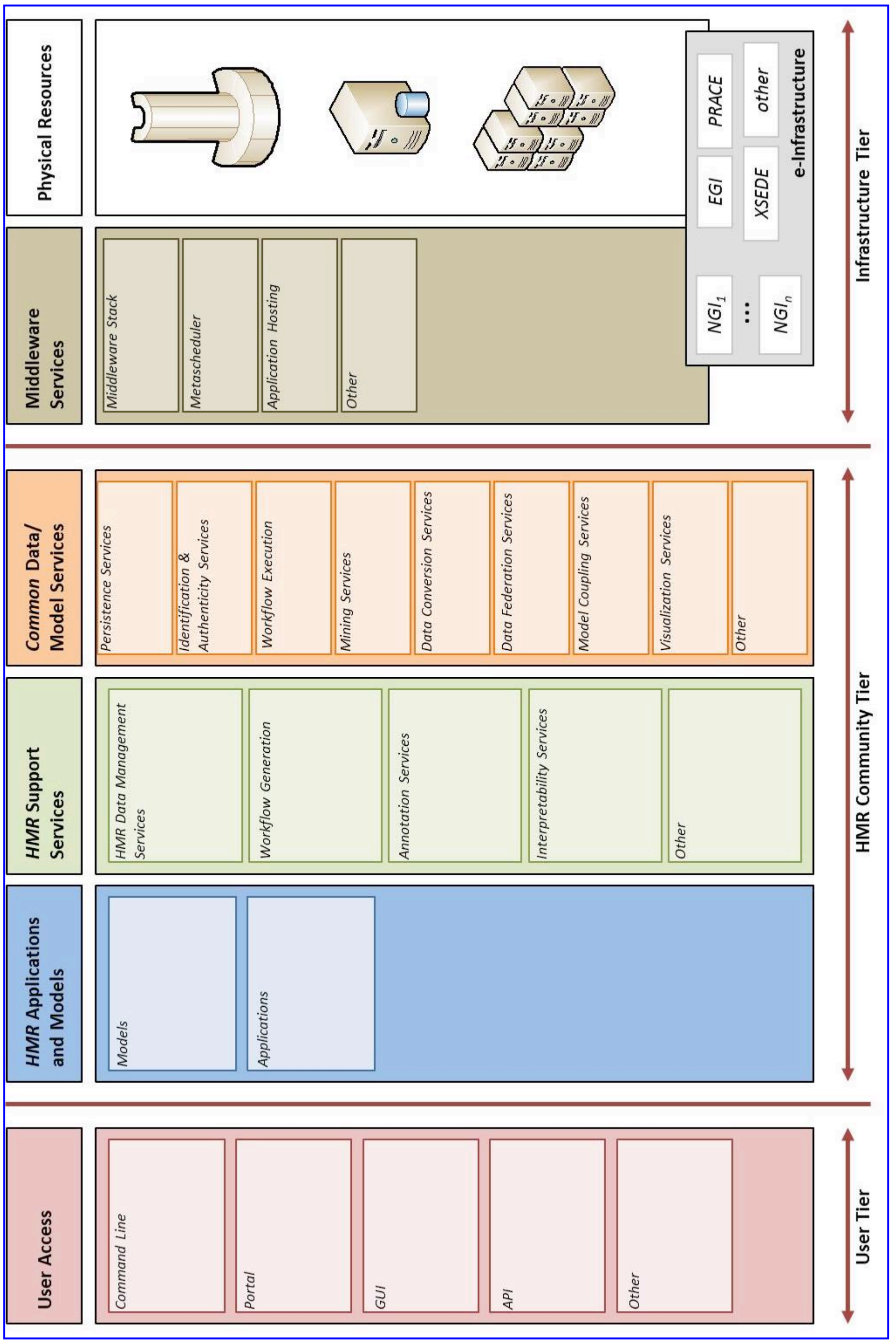

Figure 4: Generic classification scheme

In [3] we used this scheme to analyze various approaches to HMR e-infrastructures.

From an MDA perspective, the scheme in Figure 2 relates to the HMR domain model as it provides a high-level conceptual model which defines and relates “physical” and abstract objects in the HMR area of interest. In [3] it has been used to document relationships between and responsibilities of conceptual building blocks (that is, blocks that represent the concept of a group of things rather than blocks that define a programming or implementation object).

4.2 The HMR Platform Independent Model (PIM)

In the following section we specify the models for a generic HMR Science System (HSS) to define the common architecture model for HMR / Earth Sciences. We start with specifying an HMR Domain Model to document relationships between and responsibilities of conceptual classes.

4.2.1 Domain Model

The *HMR Domain Model* is a high-level conceptual model, defining the physical and abstract objects in the HMR area of interest. The model can be used to document relationships between and responsibilities of *conceptual* classes rather than *programming* classes. The HMR domain model thus shows the physical and organizational units of the HMR scientific domain and their relationships. Note that such relationships may have a multiplicity (for example, an HMR application can be assigned to *no* workflow, *one* workflow or *many* workflows). For convenience and better readability, however, we omit multiplicity annotations.

The basic HMR domain model is depicted in Figure 5. It sets the context for all subsequent considerations. In particular the model shows:

1. An HSS as we consider it in this report, will host HMR applications on an infrastructure to be used and provided by scientists.
2. Scientists may be “professional” scientists or citizen ones.

3. Infrastructures are not prescribed. They may be Grid infrastructures (with heterogeneous middleware stacks), Cloud infrastructures or classic HPC infrastructures (as for example exhibited by PRACE[5]).
4. HMR applications may participate in HMR workflows and they may use HMR science services. The services may themselves be common science services or HMR specific ones (see [3]).
5. In addition, services, workflows and applications may use data and models.
6. All these entities are described by metadata in respective catalogs.

The HMR domain model of provides now the vocabulary on which rules and other objects can be modeled.

[5] http://www.prace-ri.eu

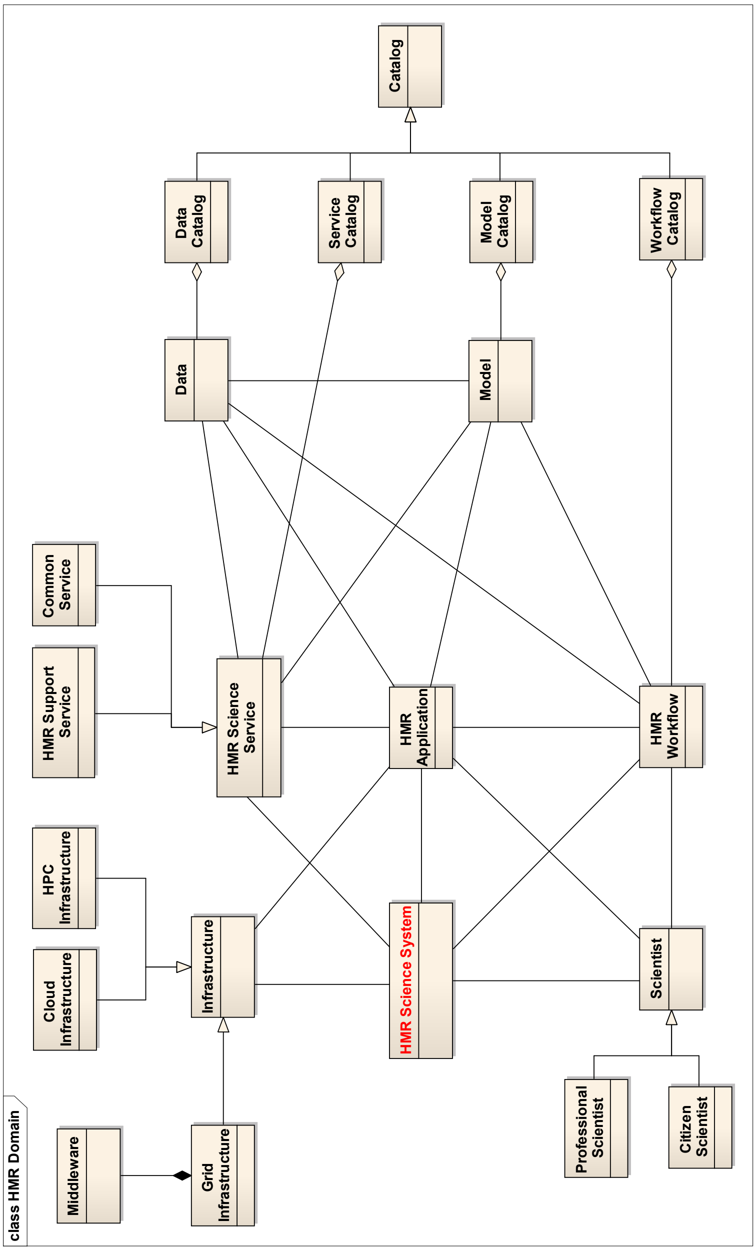

Figure 5: HMR Domain Model

4.2.2 HMR Use Case Model and Related Requirements

The HMR Use Case Model describes the proposed functionality of HSS from a usage perspective. Each HMR Use Case represents a discrete unit of interaction between a user (human or machine) and the system. It describes the functionality to be built in HSS, which can include another HMR Use Case's functionality or extend another HMR Use Case with its own behavior.

Use Case descriptions typically include general comments and notes describing the use case, formal functional requirements of things that the Use Case must provide to the end user, formal rules and limitations the Use Case operates under, invariants that must always be true throughout the time the Use Case operates, formal descriptions of the steps taken to carry out the Use Case, and any additional attributes which may be of importance. Because we do not want to go beyond the scope of this report, these formal use case descriptions are omitted here, they will be delivered in respective addenda to this report.

HMR Use Cases are related to "HMR actors", which may be human or machine entities that use or interact with HSS to perform a piece of meaningful work that helps them to achieve their scientific goals.[6]

For the scope of this report we can identify the following actors:

1. The *Scientist* which includes the *Citizen Scientist*.
2. A *Workflow Provider* to create and maintain scientific HMR workflows.
3. A *Model Provider* to create and maintain HMR models.
4. A *Data Provider* to create and maintain HMR data sets.
5. An *Infrastructure Provider* to setup, maintain and manage HMR e-infrastructures.

---

[6] The set of HMR Use Cases HMR actors have access to, defines their overall role in HSS and the scope of their actions.

6. An *Administrator* to manage access to HMR e-infrastructures.

Figure 6 shows the relationship between these actors. Please note the use case precedencies (workflows require models and data), inclusions (managing infrastructure also means managing catalogs) and the deployment stereotypes (application and models need to be deployed on infrastructures).

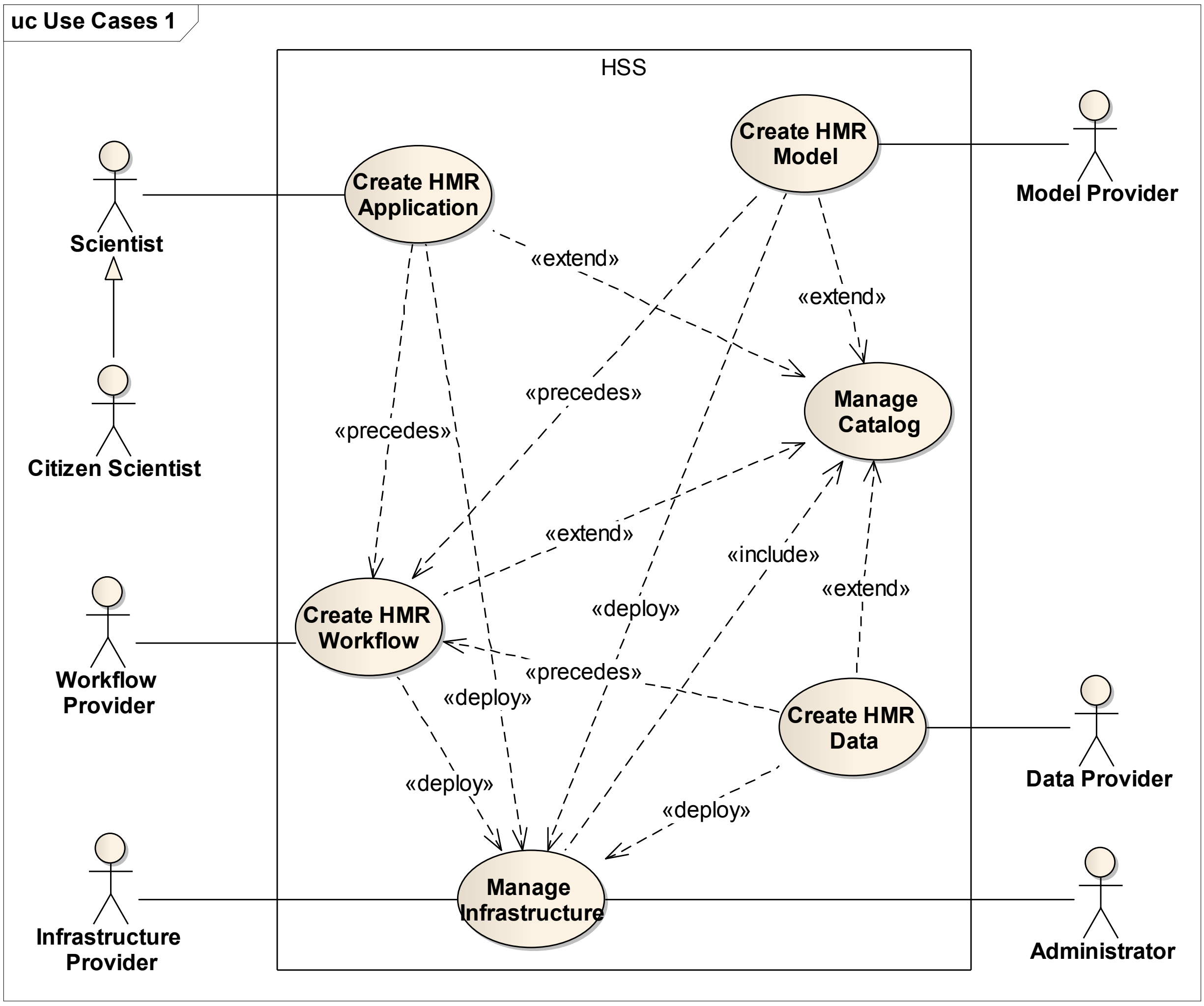

Figure 6: Use Cases for Creation and Deployment

While the use case model in Figure 6 relates to the creation and deployment of workflows, applications, models and data, the use case model in Figure 7 refers to the execution of HMR processes.

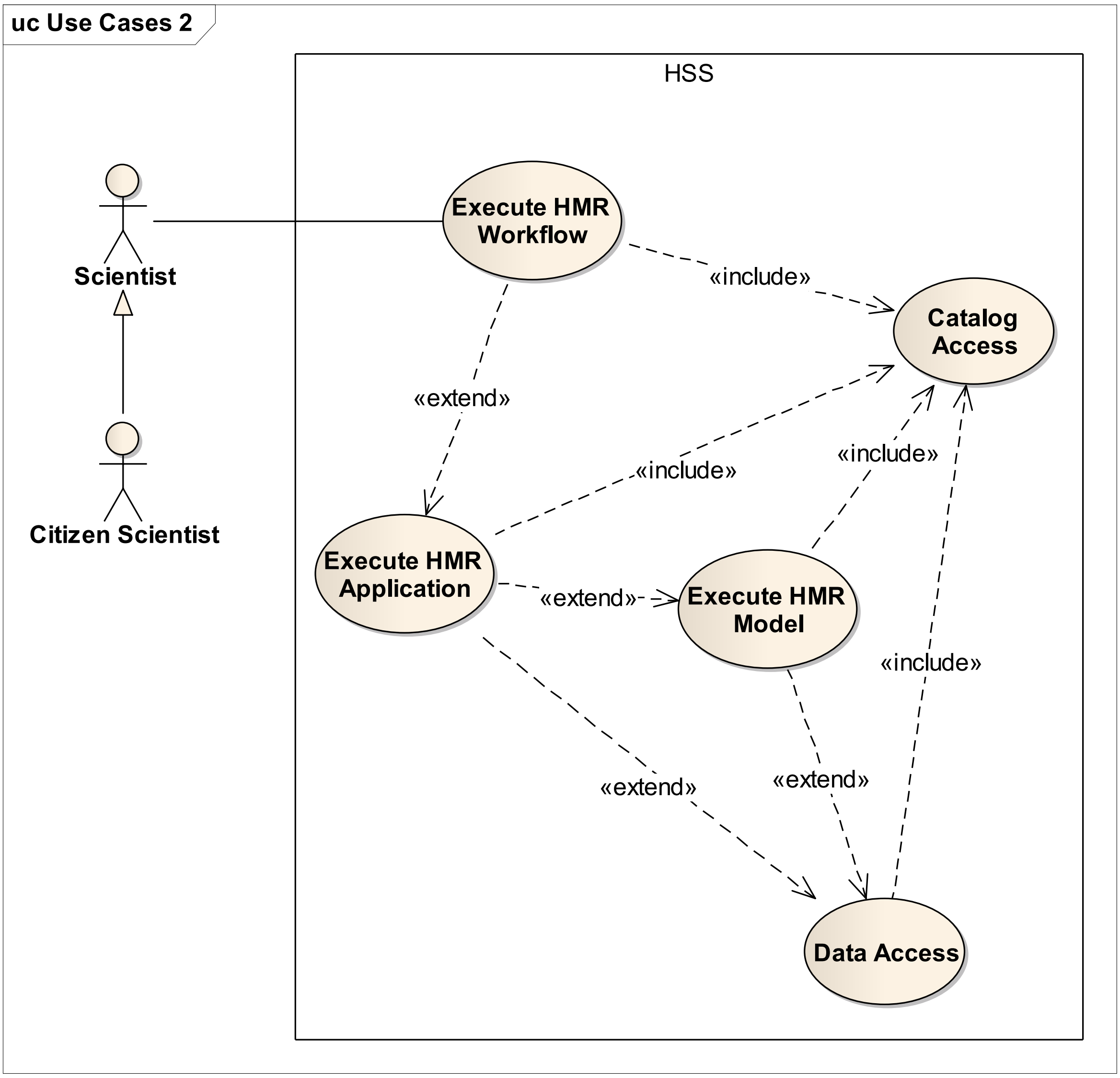

Figure 7: Use Cases for Execution

Please note again the extensions (executing HMR applications also means executing models) and the inclusions (data access requires catalog access).

There are several other use case scenarios which are omitted here for convenience because they are obvious. Examples are more detailed versions of the uses cases already depicted, uses cases referring to educational aspects, use cases for dissemination, standardization and support, or use cases for interpretability services and visualization (see Figure 4 and [3]).

Altogether the use case models describe the boundary and interaction between HSS and the various users/stakeholders. In some respect the models thus define the requirements to be satisfied.

4.2.3 Logical Model

The logical HSS model represents a static view of the objects and classes that make up the design/analysis space for HSS. While the domain model in section 4.2.1 is a looser, high level view of HSS Objects and entities, the logical class model is a more rigorous and design focused model. Figure 8 shows an example of a class diagram that relates catalog entries to models, data and workflows. For readability we omit the operations and just show the attributes.

There are several ways to model the objects and their relationships. In the approach we chose for this report we model conversions as separate entities which describe model couplings (conversion between models) and data format bridging (conversion between data). Workflow conversions are not applicable. For data conversion we propose to encode the mechanisms in the catalog entries. Catalogs are inheriting from the abstract `Catalog` class. Note that the specification in Figure 8 allows for data provenance considerations.

A coarse-grained infrastructure model is proposed in Figure 9.

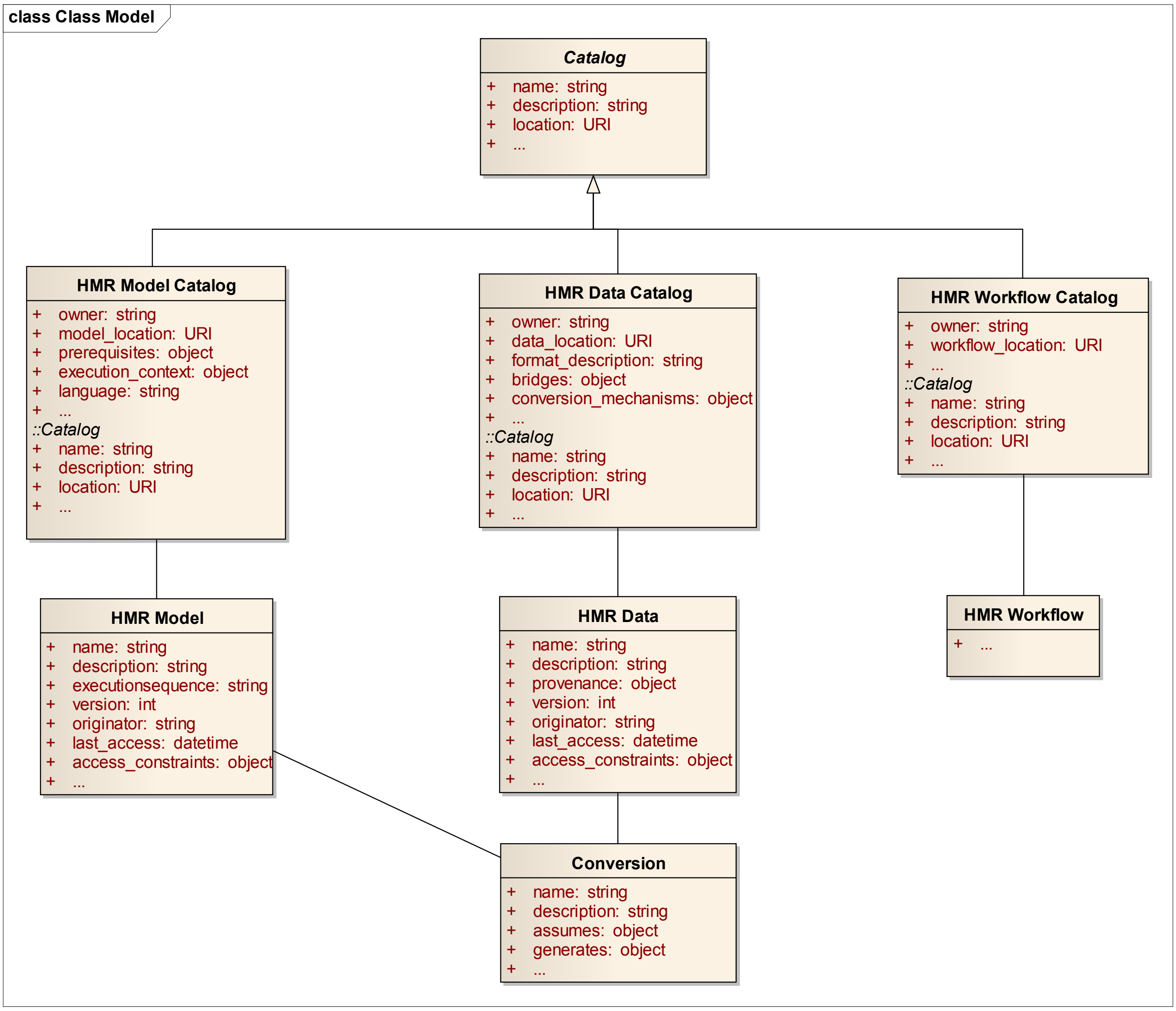

Figure 8: Catalogs and Conversion Mechanisms

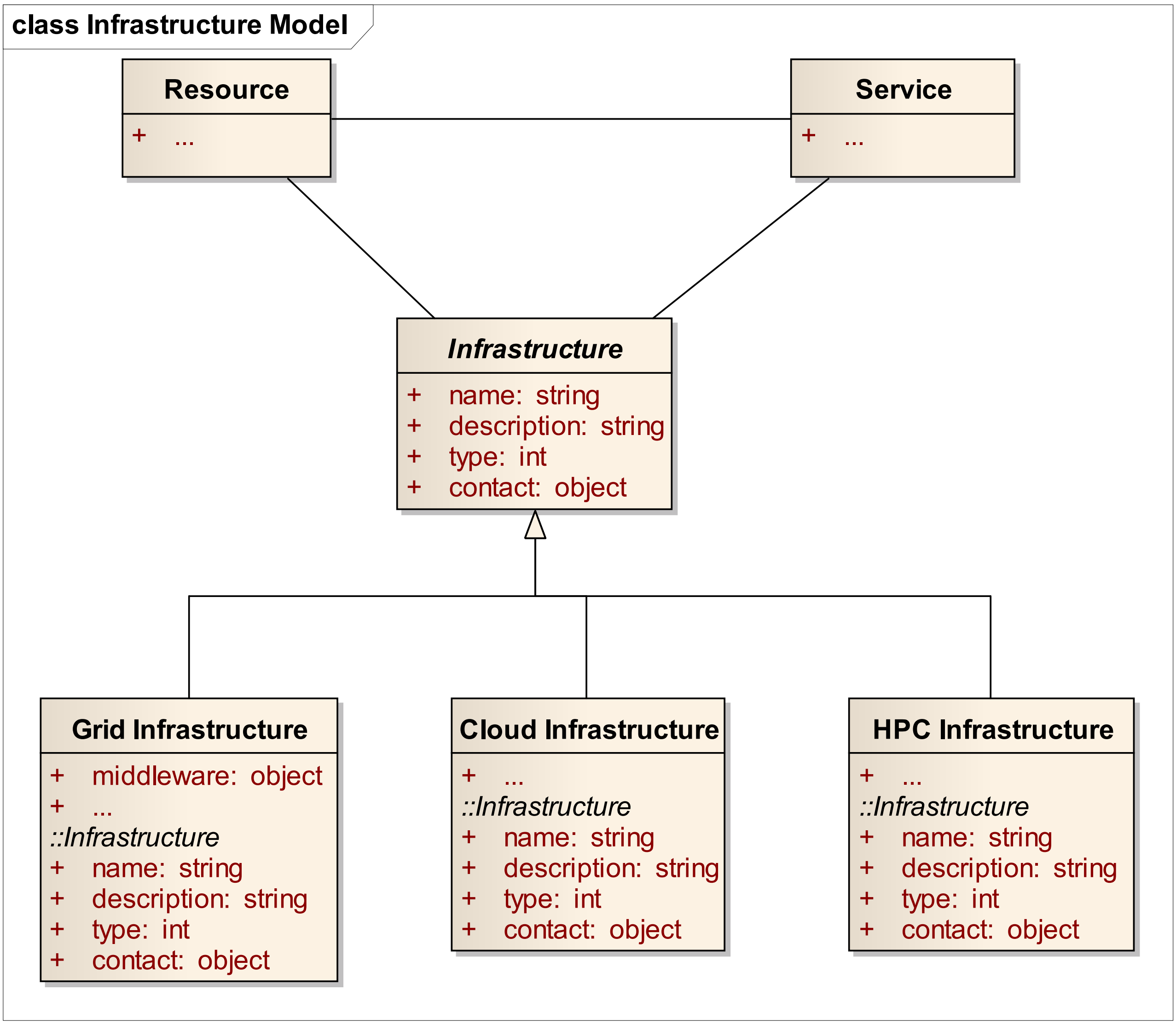

Figure 9: Infrastructure Model

Similar class specifications apply to the other domain objects shown of Figure 5. We will provide their final versions in an addendum to this report later.

4.2.4 Dynamic Model

The dynamic model is used to express and model the behavior of HSS over time. A formal specification of a complete dynamic model would go beyond the scope of this report. Rather, we informally describe a typical HMR application usage.

A typical HMR application consists of (see also Figure 10 and [12]):

- software modules simulating certain phenomena in certain time or space scale (scaleful). Usually these modules are computationally intensive and would require HPC resources. They are often (but not always) implemented as parallel programs.
- software modules that prepare data for scaleful modules and process data from scaleful modules. Usually these modules do not have demanding computational requirements. However, to avoid communication overhead, they often require to be executed "close" to the scaleful modules they are connecting (they could even be implemented in the same process as the one for the scaleful modules).

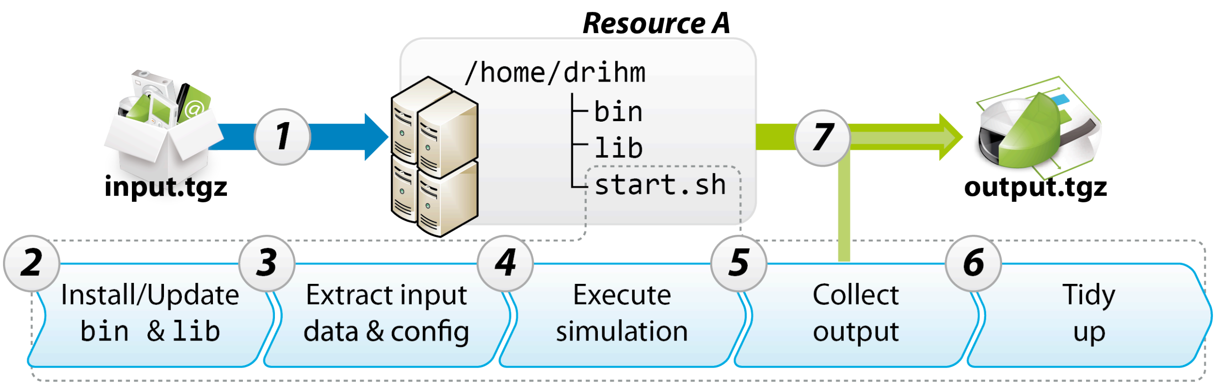

Figure 10: Typical HMR Application (DRIHM Example)

The communication structures of HMR applications may vary significantly between:

- a master-slave paradigm, where a master module triggers the execution of subordinate ones,

- a peer-to-peer type of computation where all modules are executed concurrently and exchange data in a usually asynchronous fashion. During the course of their execution, applications often pass many synchronization points (the number can be static or dynamic). Therefore, this type often requires mechanisms for efficient communication.
- a pipe type communication where modules execute one after another.
- a hybrid communication with combinations of two or more types mentioned above. The initial condition module is connected to the rest of the simulation via “pipe”, and then the rest of the simulation consists of modules that run concurrently.

4.2.5 Physical Model

The Physical/Deployment Model provides a model of the way components will be deployed across some infrastructure. Because it details network capabilities, server specifications, hardware requirements and other information related to deploying the proposed system, we will not discuss this model any further as it requires knowledge of where components will be located, on what servers, machines or hardware.

5 Conclusion

In this deliverable we provided the outline of a draft specification of a common architecture model. The specification followed the Model Driven Architecture approach and based on the findings of report D2.1.

In particular we outlined an HMR domain model, a use case model and the static class model. Both the dynamic interaction models and the physical deployment model are omitted in this report as they need further decision making. These models will therefore be provided later as an addendum to this report.

6 Acronyms and References

6.1 Acronyms and Abbreviations

Acronym / Abbreviation	Definition
API	Application Programming Interface
DDL	Data Definition Language
DRIHM	Distributed Research Infrastructure for Hydro-Meteorology
DRIHM2US	Distributed Research Infrastructure for Hydro-Meteorology to United State of America
EGI	European Grid Infrastructure
EJB	Enterprise JavaBean
GUI	Graphical User Interface
HMR	Hydro-Meteorological Research
HPC	High Performance Computing
HSS	HMR Science System
ICT	Information and Communications Technology
MDA	Model Driven Architecture
NGI	National Grid Initiative
OMG	Object Management Group
PARADE	Partnership for Accessing Data in Europe
PIM	Platform Independent Model
PRACE	Partnership for Advanced Computing in Europe
PSM	Platform Specific Model
UML	Unified Modelling Language
XSEDE	Extreme Science and Engineering Discovery Environment

6.2 References

[1] Parodi, A. et al.: DRIHM Description of Work (DoW), 2011.

[2] United Nations Office for Disaster Risk Reduction (UNISDR): From Shared Risk to Shared Value –The Business Case for Disaster Risk Reduction. Global Assessment Report on Disaster Risk Reduction. Geneva, Switzerland, 2013

[3] DRIHM2US Project: Report on an Assessment of Current e-Infrastructure Approaches for Hydro-Meteo Research in Europe and the US. Deliverable D2.1, August 2013

[4] Foster, Ian ; Kesselman, Carl ; Tuecke, Steven: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. In: International Journal of High Performance Computing Applications 15 (2001), Nr. 3, S. 200-222

[5] Wolfgang Gentzsch ; Burak Yenier: The UberCloud HPC Experiment: Compendium of Case Studies. http://www.hpcwire.com/whitepapers/2013-06-25/the_ubercloud_hpc_experiment:_compendium_of_case_studies.html

[6] Rosette Vandenbroucke: e-IRG White Paper 2011; 2011, http://www.e-irg.eu/images/stories/e_irg_whitepaper_and_comments_2011.zip

[7] Carnegie Group of G8 + 05 Science Advisers: G8+O5 GLOBAL RESEARCH INFRASTRUCTURE, SUB GROUP ON DATA, Draft Report, 2011, http://cordis.europa.eu/fp7/ict/e-infrastructure/docs/g8.pdf

[8] European Commission: The Second European Union-Australia Workshop on Research Infrastructure, Brussels, 2012; http://ec.europa.eu/research/infrastructures/pdf/2nd-aus-eu-ri-workshopreport-oct.pdf#view=fit&pagemode=none

[9] European Commission: Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions; Horizon 2020 – The Framework Programme for Research and Innovation, Brussels, 2011

[10] Sergio Andreozzi: EGI Feedback on D3 –Final Study Report (draft) "Developing impact measures for e-Infrastructures". RI-Impact –Dissemination Workshop, 2012, http://www.ri-impact.eu/ri-impact-wAssets/docs/EGI-Perspective.pdf

[11] Wilkins-Diehr, Nancy, Dennis Gannon, Gerhard Klimeck, Scott Oster and Sudhakar Pamidighantam: TeraGrid science gateways and their impact on science. Computer, 41(11):32{41, 2008.

[12] DRIHM2US Project: Report on an Assessment of Options for an Organizational Setup. Deliverable D5.2, October 2013

[13] High Level Expert Group on Scientific Data: Riding the wave – How Europe can gain from the rising tide of scientific data; Final report of the High level Expert Group on Scientific Data, October 2010

[14] Franzoni, Chiara and Sauermann, Henry: Crowd Science: The Organization of Scientific Research in Open Collaborative Projects (April 8, 2013). Research Policy, http://dx.doi.org/10.2139/ssrn.2167538

[15] PARADE: The 2009 White Paper "Strategy for a European Data Infrastructure" of the European data initiative PARADE (Partnership for Accessing Data in Europe), 2009

[16] Felix Bachmann, Ian Foster, Andrew Grimshaw, Dave Lifka, Morris Riedel, Steve Tuecke: XSEDE Architecture, Level 1 and 2 Decomposition, February 2012

[17] David S. Frankel: Model Driven Architecture (OMG): Applying MDA to Enterprise Computing; Wiley, 2010

[18] Dan Pilone, Neil Pitman: UML 2.0 in a Nutshell; O'Reilly, 2005