



# ***SIMDAT***

Data Grids for Process and Product Development using Numerical Simulation  
and Knowledge Discovery

Project no.: 511438

Grid-based Systems for solving complex problems – IST Call 2  
Integrated project



## **Deliverable**

***D12.3.3 Final Grid/DDA infrastructure  
&  
D12.3.2 Virtual Organization Final Implementation***

<b>Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)</b>		
<b>Dissemination level</b>		
<b>PU</b>	Public	X
<b>PP</b>	Restricted to other programme participant (including the Commission Services)	
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission Services)	
<b>CO</b>	Confidential, only for members of the consortium (including the Commission Services)	

---

## ***Revision history***

<b>Date</b>	<b>Version</b>	<b>Author</b>	<b>Modification</b>
05.05.08	0.1	Marta Gutierrez	Template Created
16.05.08	0.2	Cristina Arciniegas	Add Chapter 2
30.05.08	0.3	Marta Gutierrez	Add Chapter 4-to-End
02.06.08	0.4	Jean Pierre Aubagnac	Add Chapter 3
03.06.08	0.5	Baudouin Raoult	Editorial Changes
04.06.08	0.6	Jean Pierre Aubagnac and Cristina Arciniegas	Editorial Changes Chapter 3
24.06.08	0.7	Cristina Arciniegas	Add Hans Christian Hoppe's Comments
26.06.08	0.8	Marta Gutierrez	Review and add suggestions from TMB.
09.07.08	1.0	Cristina Arciniegas	Final Version

## ***Copyright***

Copyright © ECMWF and other members of the SIMDAT consortium, [www.simdat.org](http://www.simdat.org),  
2008

---

## *Table of contents*

1	Introduction .....	4
2	Final Grid Infrastructure.....	5
2.1	Integration with the BRIDGE - GRIA framework.....	5
2.2	The new Subscription Service.....	6
2.3	Subscription Services Components.....	6
2.4	The Subscription Metadata.....	7
2.5	The Subscription Request Messages .....	9
2.6	The New Subscription Model at the Data Repository.....	9
2.7	Customized Subscription Implementations.....	11
3	Distributed Data Access .....	12
3.1	GTS -VMC Integration .....	12
3.2	Metadata Generator Module.....	12
3.3	GTS-VMC Architecture .....	14
4	Ontology and Discovery.....	16
4.1	Final Ontology.....	16
4.2	Standardised Search Interface .....	17
5	Final Virtual Organisation.....	19
5.1	Security Architecture.....	19
5.2	Trust Domain Model .....	20
5.3	Authentication Service .....	21
5.4	Domain Authorities and Peer-to-Peer Secured Transmission.....	22
5.5	Authorisation Factories .....	22
5.6	Management Tools .....	23
6	Evaluation/Validation Report: Prototype at PM45. ....	24
6.1	Grid Infrastructure.....	24
6.1.1	Requirements Validation.....	24
6.2	Distributed Data Access .....	28
6.2.1	Requirements validation.....	29
6.3	Virtual Organisation and Ontology .....	30
7	Final Prototype .....	32
8	Conclusion.....	32
9	REFERENCES.....	33
10	Glossary.....	34
11	ANNEX I: Software releases .....	35

---

# 1 Introduction

This document represents the Deliverables *D12.3.3 Final Grid/DDA infrastructure* and *D12.3.2 Virtual Organisation Final Implementation*, as stated in *Annex 1- Description of Work* for the Meteorological Activity.

The document reviews the progress done for the reporting period PM36-PM45 in all the technology areas and describes the final infrastructure organized by the topics Grid, DDA (Distributed Data Access), Discovery, Ontology and Virtual Organisation. The last phase of the project has focused development efforts on the implementation of the Subscription Service and proving the prototype with a distribution system and a patching tool. Efforts have also made on writing the prototype technical documentation intended for end users, administrators and developers.

The Subscription Service brings operational capabilities to the system by delivering real-time data to the customer as soon as it is available to the SIMDAT infrastructure. This work in conjunction with the real-time data acquisition and publication modules constitutes the final blocks of development delivering an integrated infrastructure of operational strength.

During this phase the consolidation of the Virtual Organisation has made possible the publication of new secured datasets, available to organizations that establish out of band Trusting agreements. A description of security services deployed at the SIMDAT infrastructure consolidating the Virtual Organisation is provided.

Technical implementations and final developments in each of the technological areas are now being supported and integrated within the prototype. The document describes the evaluation and validation report of the prototype as of PM45 together with the description and final deployment plans of the prototype by the end of the project.

## 2 Final Grid Infrastructure

The developments during this last stage of the SIMDAT project have focused on enhancing the Grid infrastructure by implementing a data dissemination mechanism and subscription service. The new subscription service capitalizes on the consolidated VMC Grid infrastructure, seamlessly interfaces existing data dissemination systems, and benefits from the implementation of push technology for data delivery. As a result, the dissemination service is able to automatically dispatch meteorological data, when available, to interested users using a selected transport mechanism. Additionally the infrastructure has proven to be highly interoperable by easily integrating and providing access to model outputs from the BRIDGE project applications.

### 2.1 Integration with the BRIDGE - GRIA framework

The meteorological activity part of the EU funded BRIDGE project aims to create elaborated meteorological products from model outputs that are produced by geographically distributed applications. The BRIDGE project uses GRIA middleware to distribute input and output data on a Grid infrastructure. In this context, a scriptable interface that implements the VMC Simple Data Repository has been developed and deployed at the ECMWF node. The new BRIDGE data repository uses the VMC Grid infrastructure to expose BRIDGE resources as Grid services and handles any request to these services (see middle panel in Figure 2-1). Thus, any user with the right access credentials is able to submit a formula to a BRIDGE site via the Portal. The formula is parsed by the BRIDGE client, executed and the results are generated on the fly. Results are delivered in the format chosen by the user, and can be instantly retrieved from the Portal (pull data delivery mechanism).

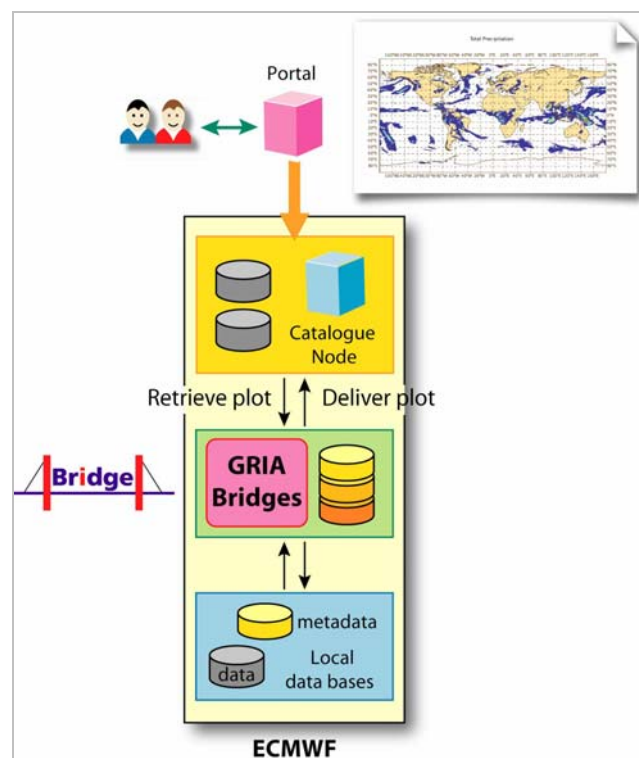


Figure 2-1 GRIA- BRIDGE Integration.

VMC infrastructure has provided to the BRIDGE applications a single point of entry where any user can submit, monitor and retrieve results from running distributed models as part of a

---

GRIA infrastructure. By leveraging the VMC infrastructure, BRIDGE applications have become accessible to any user worldwide.

## **2.2 The new Subscription Service**

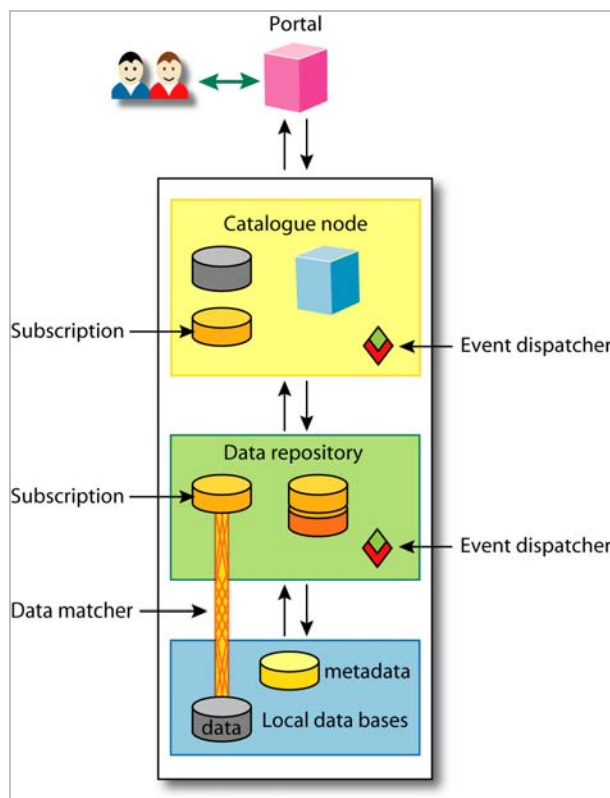
A *Subscription Service* for delivering critical data following the design presented in the [1] *D12.3.1 Service Architecture Design supporting Subscription as a first service* has been implemented. In the meteorological community, a *Subscription Service* is a data dissemination mechanism that allows the delivery of “future” data from any producer to any end user. The purpose of the VGISC Subscription Service is to offer users an efficient and reliable mechanism for receiving data that will be produced at a later stage.

From the VMC client viewpoint, a typical subscription scenario implies a user discovering via the Portal one or more data sets of interest. The user chooses a data set and parameterizes her/his subscription request by specifying time base, geographical selection, delivery mechanisms and any other parameters available for that specific dataset. If the data is “subscribe-able”, the user has the possibility to subscribe to future releases of the dataset by submitting a parameterized subscription request. The user places the subscription request and would get the customized dataset delivered automatically every time a new version is made available via the specified delivery mechanism (shopping cart, ftp, email, etc). From the VMC architectural viewpoint, every user’s subscription request is stored in a request database at the user’s main Catalogue Node and a copy is forwarded to the Data Repository holding the data. The request is routed across the mesh network to either the local Data Repository or to another Catalogue Node, if the request involves data in a remote Data Repository. In both cases, the request is stored at a Data Base in the Data Repository which owns the subscribed dataset. Thus, the Catalogue Node linked to the Portal where the subscription was placed and the Data Repository (local or remote) holding the dataset both have a copy of the subscription request.

The data delivery implemented by the on-demand request system (see [2] *D.20.1.3 Alpha version of the demonstrator focusing on the transfer of real-time data under strict security*) utilizes a pull-based mechanism that allows a user to request and retrieve at the same time a dataset. But, if the requested dataset is not available the user needs to come back at a later stage and request the data again. In contrast, at the core of the new *Subscription Service* is a push-based delivery mechanism. The implemented delivery mechanism initiates a dissemination service as soon as a new data arrival satisfies any existing subscription requests stored at the Data Repository. Thus, there is not need for users to repeatedly request the same data set. Instead, the user subscribes once and keeps receiving the dataset by the transport mechanism(s) of his/her choice.

## **2.3 Subscription Services Components**

The implementation of the new *Subscription Service* has required the addition of new functionalities to all three main VGISC components (see figure below). The Portal now renders subscription metadata (see Section 2.4) allowing users not only to parameterize their requests as usual, but also to choose multiple delivery destinations for their data subscriptions. The Portal has also been enhanced by a Web-based subscription management tool. This tool allows users to monitor the state of their submitted subscription requests (ongoing, completed or aborted), and to manage their delivery destinations. By using this tool, a user can add and delete a new email address, or configure an ftp delivery.



**Figure 2-2 New Subscription-Service Components.**

The Catalogue Node now parses the Subscription Request Messages (see Section 2.5) dispatched when a user submits a subscription request via the Portal. The Catalogue Node has been modified to maintain a subscription request database and a data delivery destination database. The former permits monitoring the users' subscription requests; the latter is used to keep information about users' data delivery destinations. A new event dispatcher enables the Catalogue Node to handle events triggered by the Data Repository, which are used to announce the matching of new data arrivals with stored subscription requests.

The re-designed Data Repository also keeps a user destination database and a subscription request database, both of which are synchronized with the equivalent databases managed by the Catalogue Node. The Data Repository has also been modified to support the new subscription metadata (see Section 2.4). An essential addition to the Data Repository is a *Data Matcher* that pairs up the new data arrivals with the stored subscription requests (see connection between the green and blue panels in Figure 2-2). The *Data Matcher* reports the new matches to the new dispatcher at the Data Repository. In turn, the Data Repository dispatcher announces the new matches to the Catalogue Node dispatcher, and starts the data dissemination process. The Data Repository has been adapted to follow through the dissemination process until it gets a confirmation from the Catalogue Node that data has been successfully delivered to the user.

## 2.4 The Subscription Metadata

The VGISC metadata scheme has been extended to include the new elements required by the Subscription Service. Thus, a *subscription services element* (<v:subscription>) has been introduced for describing the subscription information.

```

<metaData>
  <!-- VGISC INTERNAL REQUEST -->
  - <v:vgisc>
    <v:id>pt.meteo.sub.files.aladin</v:id>
    <v:inherit>int.wmo.core</v:inherit>
    <v:location>pt.meteo.sub</v:location>
    <v:resync>43200</v:resync>
  - <v:request>
    <v:class>e4</v:class>
    <v:levtype>sfc</v:levtype>
    <v:database>testdatarepository</v:database>
    <v:filename>ALADPORM_@@base_time@@_@@step_time@@</v:filename>
  </v:request>
  - <v:subscription method="subscribe">
    - <v:selection>
      - <v:base_time type="enum" multiple="1" title="Base time">
        <v:value seq="0 12 12">##h00</v:value>
        </v:base_time>
      - <v:step_time type="enum" multiple="1" title="Step time">
        <v:value start="0" inc="1" end="48">##</v:value>
        </v:step_time>
      </v:selection>
    - <v:destinations>
      <v:destination title="Shopping Cart" type="Basket"/>
      <v:destination title="FTP Delivery" type="FTP"/>
      <v:destination title="Email Delivery" type="Email"/>
    </v:destinations>
  </v:subscription>
</v:vgisc>
</metaData>

```

Parameters that will help users to tailor their subscription request regarding this dataset. These parameters are used by the *Widget Manager* of the Portal.

Delivery mechanisms that will be available for this data set. A data set can have more than one mechanism associated.

Figure 2-3 Example of the VGISC Extension Metadata with the New Subscription Element (red rectangle).

The `<v:subscription method="...">` element (see Figure 2-3) indicates which dissemination system can be used with the given dataset. Currently, the subscription service offers two data dissemination methods: pull and push (subscribe). The *pull* method, used by on-demand requests, indicates that the dataset is delivered right after the user submits the request using any of the delivery mechanism(s) indicated by the `<v:destination>` element. In contrast, the *subscribe* method implies the dataset can be regularly delivered as soon as the new versions become available. The `<v:destination>` element indicates which delivery mechanisms will be available to this dataset. The metadata in Figure 2-3 shows that the described dataset can be regularly delivered (subscribe) using the Shopping Cart at the Portal, Ftp or Email delivery. Additionally, the `<v:selection>` elements allow users to retrieve sub-selections of the data set based on other parameters such as base or step times as in the example shown in the Figure 2-3.

The subscription metadata rendered by the Portal helps users to customize their requests and to choose the data delivery mechanisms from the available implementations. This information is parsed to and used by the Catalogue Node to assemble a *subscription request message*. The Catalogue Node dispatcher is in charge of sending the subscription request message to the dispatcher at the Data Repository.

## 2.5 The Subscription Request Messages

Two new VMC messages (*DeleteSubscribeRequestMessage* and *SubscribeRequestMessage*) have been added to the VMC communication protocol to implement the subscription service. The *DeleteSubscribeRequestMessage* is used by the Dispatcher at the Catalogue Node. This message tells the Data Repository Dispatcher to delete a subscription request from the subscription request database. A *SubscribeRequestMessage* is used by the Catalogue Node Dispatcher to inform the local or remote Data Source about a new subscription request.

The *SubscribeRequestMessage* is assembled by the Node using the `<request>` and the `<subscription>` metadata elements of the dataset that has been selected by the user at the Portal. The example in figure 2.4 illustrates an assembled subscription request that uses “My Email” as the data transport system. My email will deliver the requested data using the specified email address: *Marta.Gutierrez@ecmwf.int*.

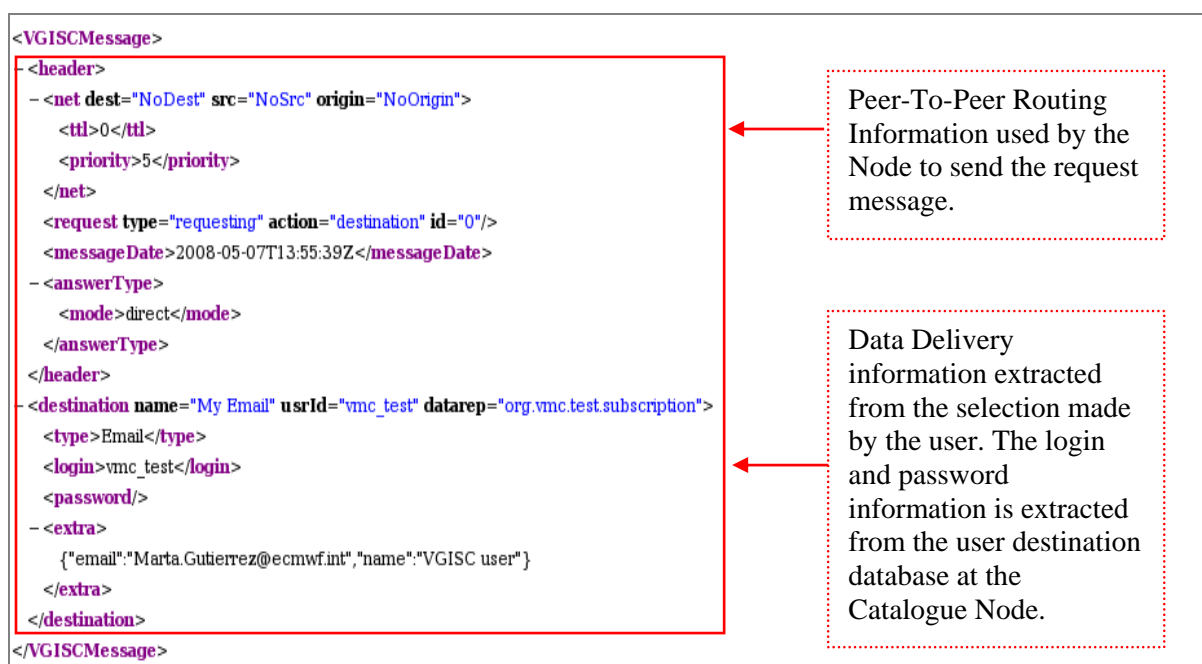


Figure 2-4 An example of a Subscription Request Message

Following the current implementation of the VMC communication protocol, ([2] *D.20.1.3. Alpha Version of the demonstrator focusing on the transfer of real-time data under strict security*), all VMC messages are mapped into Action Objects. Thus, two new Action Objects have been implemented to support the new functionalities involved when a subscription request is submitted (*SubscribeRequestMessage*) or deleted (*DeleteSubscribeRequestMessage*).

## 2.6 The New Subscription Model at the Data Repository

The Data Repository has been re-designed to support all changes demanded by the new subscription service ([1] *D.12.3.1 Service Architecture Designs supporting Subscription as a first service*). The Data Repository needs now to orchestrate the work between the pull/push dissemination systems and local meteorological databases. The new Data Repository design comprises three elements: the *Data Source*, the *Data Transport* and the *Data Requester*.

- **Data Source**, implemented as a thread, reports the arrival of new data to the Catalogue Node by using a Data Matcher that pairs up new data arrival with the subscription requests.
- **Data Requester** retrieves the requested data following the *SubscriptionRequest* Message parameters from the designated staging area. The Data Requester keeps the Data Transport informed about the location (path file name) of the resulting file.
- **Data Transport** delivers the final data file to the user using the selected transport method(s) extracted from *SubscriptionRequestMessage*.

The following figure illustrates the interactions among all VMC components when a user submits a subscription request and when the data matching the user's subscription arrives.

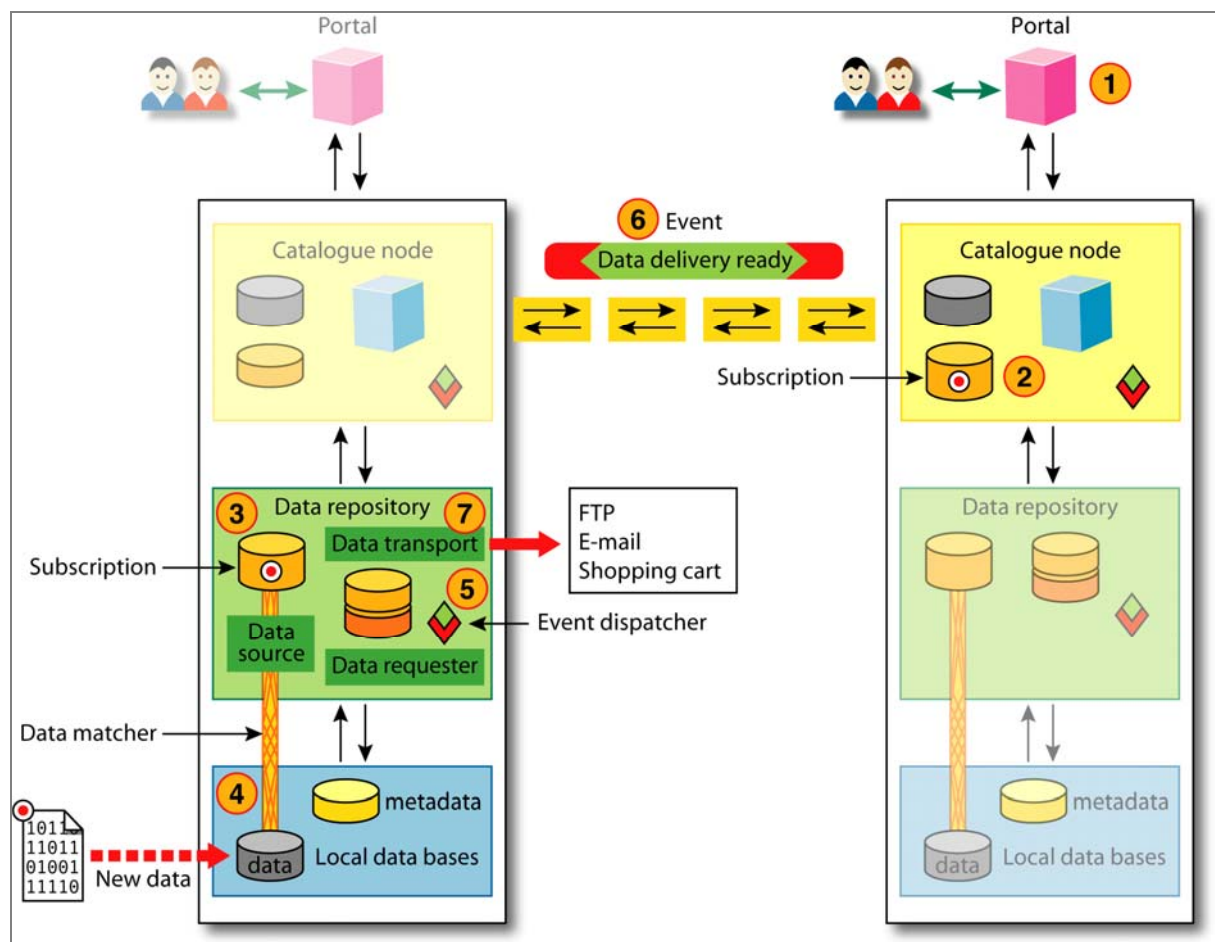


Figure 2-5 The VMC Subscription Service Architecture.

1. A user at the Portal submits a subscription request to a dataset in a remote Data Repository (left hand green panel).
2. The subscription request is stored at the local Catalogue Node subscription database (right hand yellow panel). The Node Dispatcher (red-green diamond) sends a *SubscribeRequestMessage* informing the Catalogue Node in charge of the remote Data Repository.
3. The subscription request is stored in the subscription database at the remote Data Repository (left hand green panel).
4. New data arrives at the remote databases (left hand blue panel). The data matcher checks the new arrival against the subscription requests stored at the local database.

5. If there is a match, the Data Requester assembles the final data file following the parameters defined at the *SubscribeRequestMessage*. The Data Requester also informs the Data Transport about the location of the assembled data file.
6. The Data Repository Dispatcher triggers a *DATADELIVERYREADY* event to notify the Catalogue Node were the subscription was first placed about the subscription matching. The data delivery process is initiated.
7. The Data Transport delivers the final data file to the user. It uses the delivery mechanisms specified in the *SubscribeRequestMessage*. Once all data deliveries are successful, the Data Transport triggers a *DATADELIVERYCOMPLETED* event.

## 2.7 Customized Subscription Implementations

The new Data Repository Interface organizes the interaction among the new subscription components: Data Source, Data Request and Data Transport. The new version of the VMC offers built-in implementations of the subscription components. The following table lists the implementations available for each component.

Core Component	Available Implementation	Functionality
<b>Data Source</b>	File Data Source	Detects when a new file arrives into a given directory.
	Script Data Source	A shell script that is periodically executed and creates an event file when new data is available.
<b>Data Requester</b>	File Data Requester	A module that retrieves the filename of the delivered dataset from the event file.
	Script Data Requester	Calls a script in a same way as the Zero Implementation Data Repository.
<b>Data Transport</b>	Email Data Transport	Sends the subscribed data to a specified email address.
	FTP Data Transport	Uses the file transport protocol to transfer the subscribed data.
	SFTP	Uses the Secure File Transfer protocol to transport the subscribed data.
	Shopping Cart	Delivers subscribed data to user's shopping cart at the Portal.
	Script Data Transport	Executes a script that implements the data transfers.

**Table 2-1 Implemented Built-in Services.**

If the user wants a customized implementation of the Subscription Service, the three main components (Data Source, Data Requester and Data Delivery) should be overridden as needed.

---

## 3 Distributed Data Access

The VMC infrastructure has been developed within the framework of the WMO-WIS program. The WMO-WIS is seen in the long term as the replacement for the current WMO GTS. In alignment, the VMC infrastructure has been enhanced by designing and partly implementing an interface that helps acquiring real-time meteorological data from the GTS (see **Meteo-P3.15** requirement defined at [3] *D.12.2.5 SIMDAT Meteorology Application: Evaluation/Validation report and statement of requirements for project phase III*). Thus, the interface interoperates with the current GTS and will facilitate the transitional phase between the GTS in its current form, and the WIS.

### 3.1 GTS -VMC Integration

One of the main aims of the GTS is to provide a rapid and reliable round-the-clock collection and distribution of all meteorological and related data, forecasts and alerts. A key component at the centre of the GTS is the Message Switching System (MSS). An MSS provides mechanisms to receive and forward real-time data in the form of collections of reports and observations called *bulletins*.

All meteorological bulletins transmitted as part of global, inter-regional and regional exchange are declared in the WMO N°.9 *Volume C1 Reference Catalogue*. Bulletins bound for global exchange are termed as *essential*. The scope of the VMC-GTS integration is limited to the acquisition of the *essential meteorological bulletins* listed in the Volume C1.

Bulletins are acquired from the local MSS and stored in a bulletin *Cache*. This link between the MSS and the GTS cache is bi-directional. It is the first part of the VMC-GTS integration, which also consists of an interface between the GTS Cache and the VMC. This interface enables the discovery of GTS bulletins and the retrieval of their cached instances via the Portal. Both functions require metadata records for all acquired GTS bulletins. This metadata population is regularly generated from the current WMO Volume C1 reference as well as from other sources of information.

### 3.2 Metadata Generator Module

The GTS metadata generation process is driven by the current content of the Volume C1 catalogue. Each meteorological bulletin is uniquely identified by its *Abbreviated Heading Line (AHL)*. The Volume C1 catalogue quotes the coded words termed as *designators* or *indicators* which compose the AHL, and provides additional information on the bulletin format, content and transmission time(s).

This last piece of information never documents a single occurrence of a bulletin, but the context in which the bulletin is transmitted – e.g. a list of hours for routine bulletins. As such, a bulletin is not a single individual element, but a group or class of individual bulletin instances. The GTS metadata records have the same granularity as the Volume C1 catalogue as they describe classes of products instead of product instances.

#### *Scope*

GTS Metadata Generation is limited to essential bulletins destined for a global exchange. As several actors (VMC Nodes) may generate GTS metadata records in parallel, a mechanism is needed to define the scope of each Node's contribution. Within the GTS implementation this role is done by a centralized mechanism named as the *white-list*. The white-list defines the

publishing responsibilities for each contributing Node with a set of loose regular expressions applied to the bulletin's AHL. The use of the white list assures building a consistent and comprehensive metadata catalogue.

### Schema

In order to make the bulletin instances “discoverable” and “retrievable” via the Portal, their metadata need to include two sections for the discovering and the retrieving functions respectively. The discovery section follows the version 1.0 of the WMO profile of the ISO 19115 norm. The retrieving section contains the metadata elements which enable subscribing to GTS products using both *pull* and *subscribe* VMC dissemination methods (refer to Section 2.4). Subscription requests for GTS bulletins may be restricted to one or several time slots, which possibly reflect the regular bulletin transmission hours.

The figure 3.1 shows how the main elements in the GTS metadata records relate (content similarity) to the “classical” SIMDAT metadata elements.

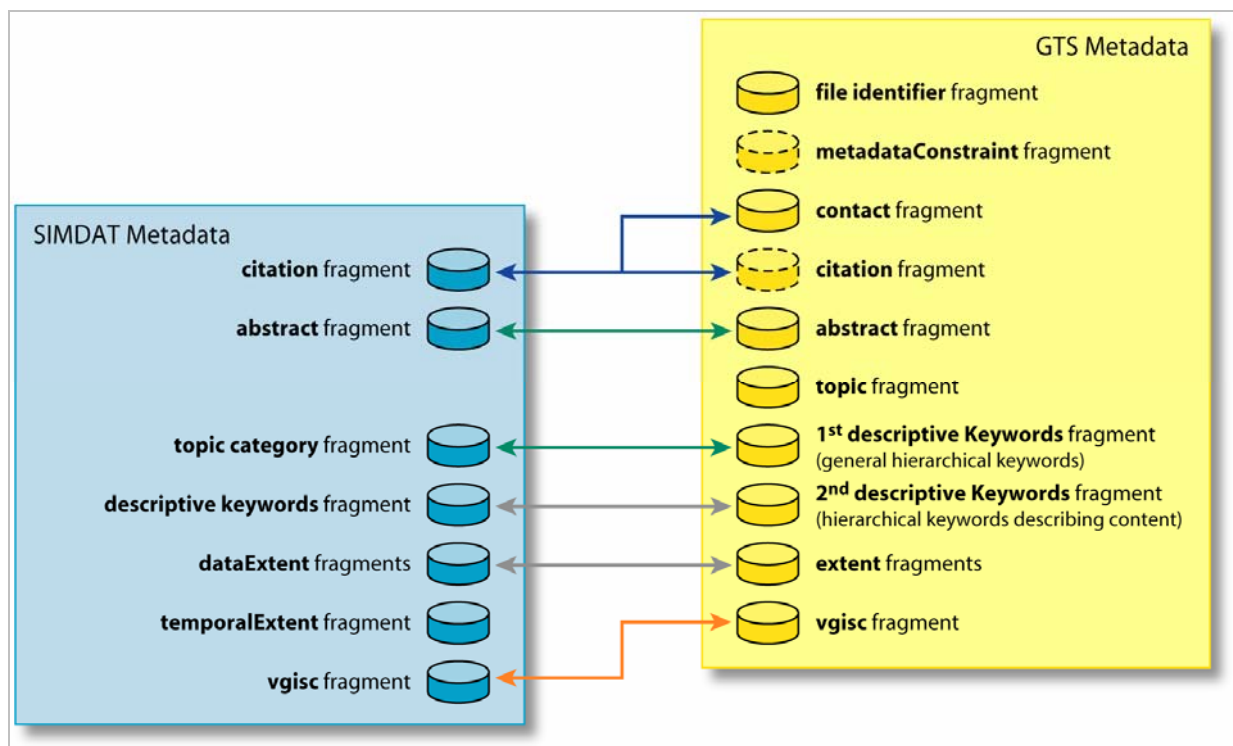


Figure 3-1 Mapping between the GTS Metadata main elements (yellow panel) and the “classical” SIMDAT metadata elements (blue panel). Bi-directional arrows indicate similar contents between the two metadata schemas.

### Information Classes

The need for information is guided by the intention to provide as much or more information in the metadata record of a bulletin as in its *Volume CI* declaration. One identified need is the expansion of coded information, designators or indicators, in order to make the metadata more explicit and not just fit for an expert eye. The need for information is also determined by the need to populate the core metadata elements following the ISO 19115 norm. Lastly, it is also determined by the need to sort, classify and tag the products for discovery via the Portal.

A set of *information classes* is constructed to meet the need for information. They are either representations of available references such as WMO N° 9 Volume A, sections of WMO N° 306 Manual on Codes or of WMO N° 386 Manual on the GTS, information collected from

sparse WMO sources, or information created for the purpose. They are implemented as XML files or XSLT style sheets. The information classes need to be centralized to ensure the content homogeneity of the metadata records created at different nodes, as well as to facilitate their maintenance.

### Operation

The *Metadata Generator* runs as an independent module at the Data Repository level in the VMC topology. Regular operation is needed as the GTS metadata population needs to evolve with WMO Volume C1, the information classes and the white-list. All these three main inputs are downloaded from their respective sources at the start of the GTS Metadata generation procedure. The generated metadata records are placed in the directory harvested by the Node.

### 3.3 GTS-VMC Architecture

The GTS integration principally adds four new components to the Data Repository (see Figure 3.2). They are the *Metadata Generator*, the *GTS Ingestion Daemon*, the *VMC-GTS Retrieval Service* and the *MSS Feeding Daemon*. The Node and the Portal are mainly left unchanged. A few changes are necessary in order to guide the Node through the GTS metadata records for indexing. The new GTS Metadata is mainly rendered by the Portal using the existing renderers.

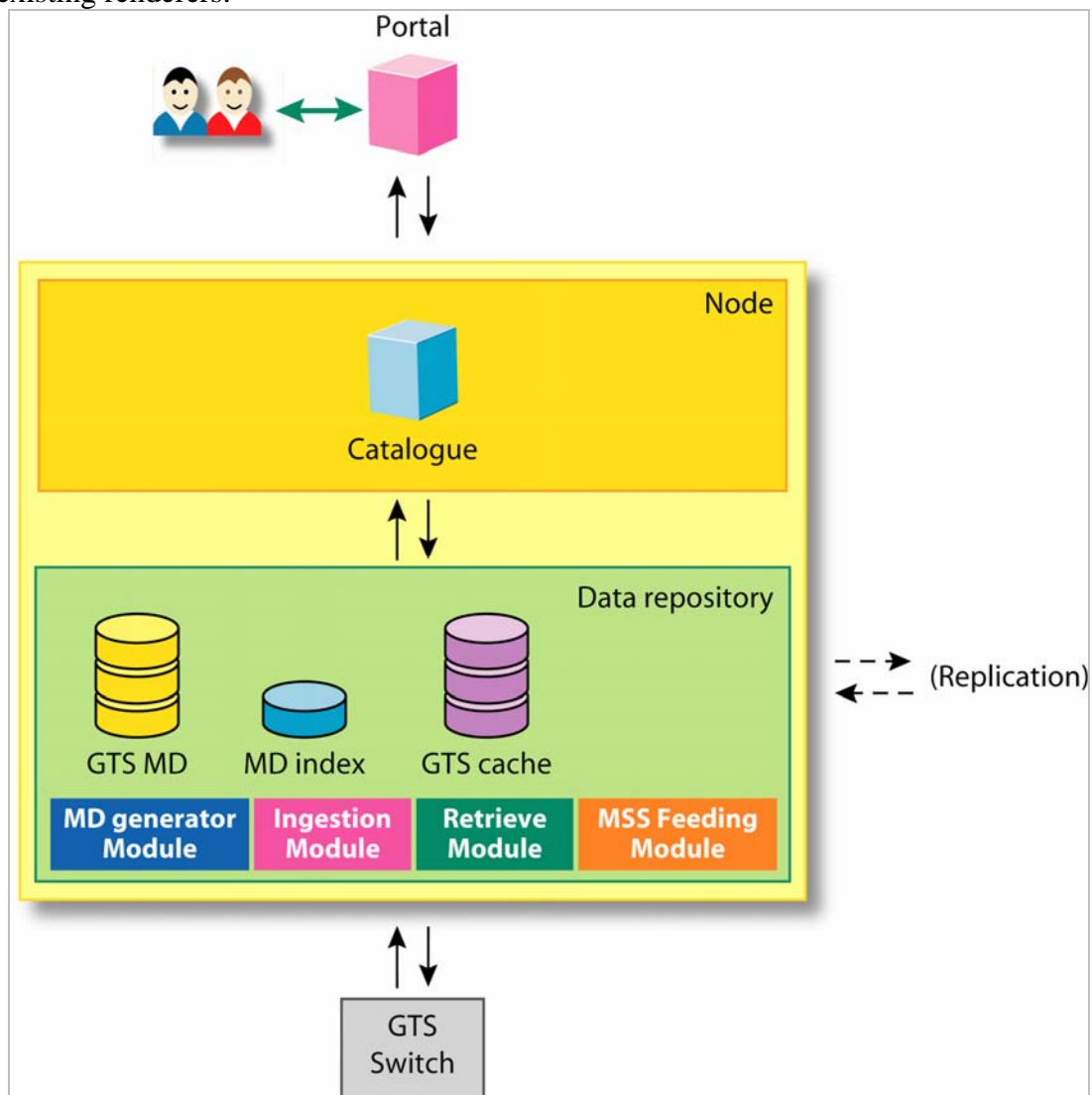


Figure 3-2 GTS-VMC Integration Architecture.

- 
- ***GTS Ingestor Daemon*** ingests information from the local MSS. The chosen strategy is to split the ingestion task among the implementing actors: a VMC Node publishing a metadata record for a GTS bulletin is assigned the responsibility to ingest the *instances* of this GTS bulletin.

The module polls the dedicated MSS output channels for incoming files. Each file packs GTS bulletin instances according to the WMO FTP Protocol (WMO N°386 Manual on the GTS – Attachment II-15). The *GTS Ingestor* filters unpacked *bulletin instances* by their AHL. A *bulletin instance* is retained only if a metadata record was locally published for the bulletin. Retained instances are inserted in the *GTS Cache*.

In the WIS context, essential GTS products will ultimately be exchanged among VMC Nodes by the *replication* process. As newly ingested products will need to be proposed to other VMC Nodes, they need to be packed with the relevant information into files and deposited in an “*outgoing for replication*” directory. Relevant information includes the bulletin identifier, the Data Repository where the associated metadata record was published, and the data repository where the instance was ingested. As a first step, a simple interface that packs comma separated pieces of info in ASCII files has been implemented. However, a fully functional replication process will be implemented outside the SIMDAT project.

- ***MSS Feeder*** is the reverse link between the GTS Cache and local MSS. Its role is to handle GTS products proposed by another VMC Node via the future replication process. As for the *GTS Ingestor*, a simple file transfer interface to the replication process is implemented.

The *MSS Feeding* module awaits the arrival of GTS bulletin instances packed in simple format files in the “*incoming from replication*” directory. Unpacked bulletin instances already part of the GTS Cache are discarded. Retained bulletin instances are inserted in the *Cache* and proposed to the local MSS. To this end, they are first sorted by WMO code form, number and time according to WMO instructions. Then, they are packed into WMO FTP Protocol files which are fed to the *local MSS*.

- ***VMC-GTS Retrieve*** extends the newly implemented Subscription Data Repository (refer to Section 2.3) and the VMC Service Interface. Communication between the Node and the GTS Data repository is done by exchanging XML messages based on the VMC communication protocol.
- ***GTS Daemons*** are developed in Java 6 and run in the background at the Data Repository level. The *GTS Data Cache* is implemented as a relational database. The *GTS Retrieval Service* is under development and the *GTS daemons* are in the testing phase.

## 4 Ontology and Discovery

The Ontology area has finalised its development demonstrating the value of Ontologies and Thesaurus in the discovery layer. Some of the latest achievements in this area have also introduced a new search interface compliant with ISO23950, a well-known search and retrieval standard widely adopted in the geospatial and library communities.

### 4.1 Final Ontology

The Ontology developments have been consolidated with the deployment of Thesaurus and Ontology services and its integration with the search engine at the Portal.

The Ontology services have been deployed at DWD exposing AMS/SKOS, GEMET and SWEET ontologies and thesaurus. A Web service client is invoked from the Portal, retrieving back the concept or keyword searched for and all the term relations extracted from the Ontology server. The search will trigger the request for the Catalogue Node trying to match metadata records and a parallel request to the Ontology server bringing back the related keywords. In this way the user will be able to identify semantically related keywords that will lead to the discovery of new datasets. Multilingual capabilities have been demonstrated through the GEMET ontology, where the keywords and its related terms are extracted via unique identifiers. In such case, a user searching for “pluie” will be able to find metadata records which contain the term “rain/pluie/lluvia” irrespectively of the language, since all these language expressions are matched by the same unique identifier.

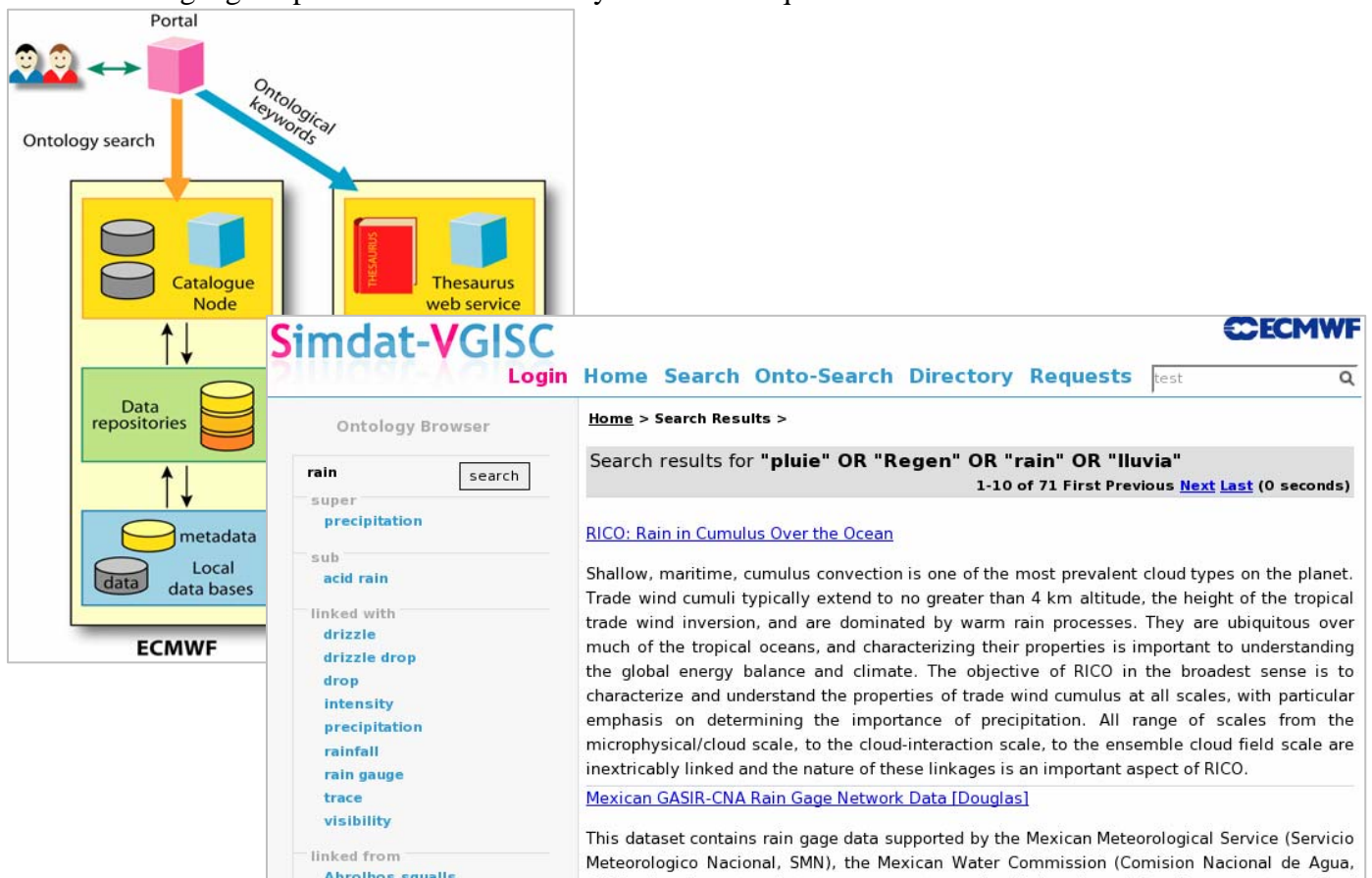


Figure 4-1 Ontology navigation from the side menu of the Portal.

---

## 4.2 Standardised Search Interface

Looking further beyond the boundaries of the SIMDAT Meteo activity and the present prototype deployment, the VMC software has been reviewed for interoperability with other WMO programmes and other communities. As of today there is widespread usage of the standard ISO23950, which standardises information retrieval syntax for Internet exposed databases. It was initially focused on the library communities but soon extended to others such as the geospatial and therefore meteorological world.

The latest developments providing enhanced interoperability have been made on the discovery service towards the implementation of an ISO23950 compliant interface.

The present SIMDAT Meteo Web interface offers the user the capability for discovering datasets through the search engine that queries the catalogue of metadata records. The searches implemented so far are keywords, temporal and geospatial queries. The way of querying the present catalogue was soon abstracted from the underlying metadata format, identifying configurable objects that are searchable and therefore indexed. Thanks to this flexible approach the search interface was easily extended to support queries compliant with ISO23950.

The SIMDAT Portal component has implemented the SRU (Search/Retrieve URL Service) protocol, the REST-full Web service version of the protocol (as opposed to the SRW, based on SOAP). This particular version is not only supported by the geospatial or library communities, but also players like Google or Yahoo offer a REST-full interface to query their Internet exposed databases.

The SIMDAT Portal accepts queries conforming to the standard ISO23950 sent in a well constructed URL. The main operation supported is the *searchRetrieve*:

[http://server/search\\_results\\_sru?operation=searchRetrieve&version=1.1&query=ozone](http://server/search_results_sru?operation=searchRetrieve&version=1.1&query=ozone)

---

The server implements the correct diagnosis messages for those operations not implemented or supported. Query parameters and values are passed to a CQL (Common Query Language) parser, which constructs checks and validates the grammar. This validated grammar expression is then mapped to the SIMDAT internal objects that will query the metadata catalogue. The search engine will respond back ResultSet Objects that are again expressed in the syntax compliant with the standard.

```
- <searchRetrieveResponse>
  <version>1.1</version>
  <numberOfRecords>1</numberOfRecords>
- <records>
  - <record>
    <recordSchema>info:sru/schema/wmo</recordSchema>
    <recordPacking>xml</recordPacking>
  - <recordData>
    - <metaData>
      - <identificationInfo>
        - <citation>
          - <resourceTitle>
            Mid tropospheric flow and boundary layer temperature
          </resourceTitle>
        </citation>
      </identificationInfo>
    </metaData>
  </recordData>
  </record>
</records>
</searchRetrieveResponse>
```

Figure 4-1: XML Response from SRU compliant SIMDAT server.

---

## 5 Final Virtual Organisation

The Virtual Organisation has been one the main block of developments of the last phase of the project. The Virtual Meteorological Centre provides the components necessary to access distributed datasets in a secured and controlled manner. The security model has been implemented in a fully distributed environment, where out of band trust relationships need to be established in order to ensure secured access.

### 5.1 Security Architecture

The VMC architecture has included a security model capable of accommodating some of the requirements that were covered by the Grid infrastructure plus other requirements that came out of the support for existing security mechanisms and policies. An extended description of the main concepts behind the security model is provided at [4] *D12.2.2 VO Architecture* and [5] *D12.2.3 Documentation on the new version of the demonstrator supporting VO functions: management of users and data policy and interoperability*.

To summarise, the security components implemented within the VMC added support for:

- Pluggable authentication and authorisation schemes: The VMC provides an integrated approach to distributed data repositories where existing schemes to authenticate and authorise users must be supported.
- Secured VMC messages: The VMC is based on a message protocol where XML messages are interchanged between the different components of the infrastructure through a REST/SOAP like interface. Thus, the technology and implementation of the security architecture needs to be aware of XML encryption and XML digital signature.
- Peer-to-peer distribution of messages: VMC messages travel through a distributed infrastructure where messages are routed through the shortest path between any two nodes of the “mesh network”. Well defined boundaries of trusted nodes are mandatory to fulfil this requirement.
- Maintain users’ privacy: user private details must be kept within the organisation in charge of registering its own users: when a user requests data located at a remote organisation, the user’s personal details must never leave the “home site” where he/she is registered. The remote location would then process a request from an authenticated user of a trusted site, but know no other details about the particular user.
- Provide support for existing data access policies (e.g. WMO Resolution 40, which is recognised globally across the WMO data centres) and also for new data policies that are applicable to a specific group.

The following figure represents the security services developed within the VMC architecture in order to support the Virtual Organisation.

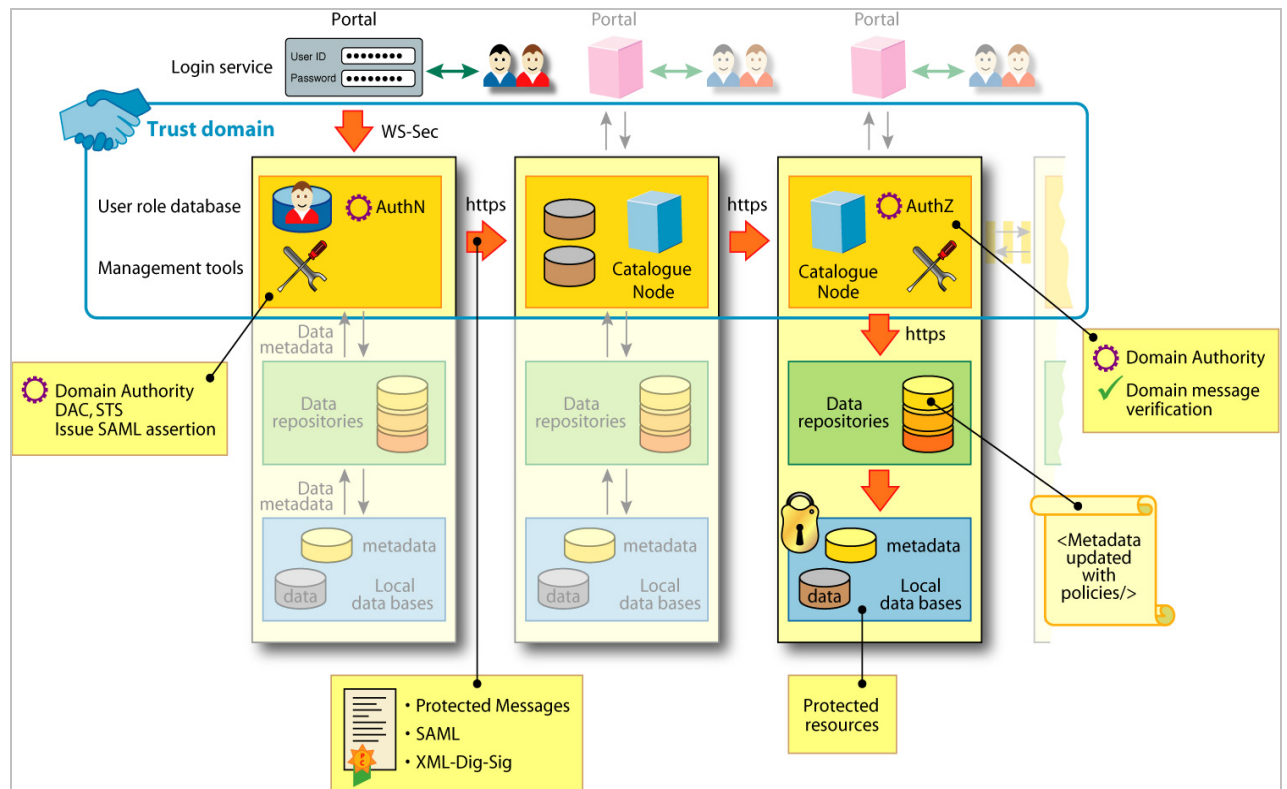


Figure 5-1: Security Components added to the VMC infrastructure.

## 5.2 Trust Domain Model

Trusted Domains are introduced in the VMC architecture in order to define a set of organisations that agree on a set of policies. This grouping could encompass two organisations, establishing a bilateral agreement, or a higher number. A domain establishes a special relation between distributed nodes that trust each other in order to serve requests originated by an authenticated user at any of the trusted nodes of the domain.

Trust relation is implemented technically through:

- Exchange of public keys between all the nodes of a Domain
- Creation of the Domain group.
- Creation of Domain policies, where policies are expressed as roles or attributes to define data access and they are distinguished between domains by a qualifying prefix that uniquely identifies the domain. Domain policies will be expressed in the metadata files describing a dataset. When policies match those assigned to a user, access to the dataset will be granted, provided that Domain membership condition is previously satisfied (see Section 5.5 on Authorisation Factories)

As of today, given the fully distributed nature of the VMC nodes, the recognition of a Domain group and its policies is a non automated practice. This information has to be imported equally in each of the nodes forming a Domain.

### 5.3 Authentication Service

An authentication service has been implemented in the Node component and it talks in the background to a user, role and domain database. The authentication service is exposed through a Web service interface in the Node.

```
-<wsdl:portType name="SimpleSecLoginServicePortType">
  -<wsdl:operation name="loginWithToken">
    <wsdl:input name="loginWithTokenRequest" message="tns:loginWithTokenRequest"> </wsdl:input>
    <wsdl:output name="loginWithTokenResponse" message="tns:loginWithTokenResponse"> </wsdl:output>
    <wsdl:fault name="NodeServiceException" message="tns:NodeServiceException"> </wsdl:fault>
  </wsdl:operation>
  -<wsdl:operation name="loginWithCertificate">
    <wsdl:input name="loginWithCertificateRequest" message="tns:loginWithCertificateRequest"> </wsdl:input>
    <wsdl:output name="loginWithCertificateResponse" message="tns:loginWithCertificateResponse"> </wsdl:output>
    <wsdl:fault name="NodeServiceException" message="tns:NodeServiceException"> </wsdl:fault>
  </wsdl:operation>
  -<wsdl:operation name="loginWithPassword">
    <wsdl:input name="loginWithPasswordRequest" message="tns:loginWithPasswordRequest"> </wsdl:input>
    <wsdl:output name="loginWithPasswordResponse" message="tns:loginWithPasswordResponse"> </wsdl:output>
    <wsdl:fault name="NodeServiceException" message="tns:NodeServiceException"> </wsdl:fault>
  </wsdl:operation>
</wsdl:portType>
+<wsdl:binding name="SimpleSecLoginServiceHttpBinding" type="tns:SimpleSecLoginServicePortType"></wsdl:binding>
+<wsdl:service name="SimpleSecLoginService"></wsdl:service>
```

Figure 5-2: Web Service Authentication.

The service is accessed through a Web service client deployed at the Portal. The implementation supports the authentication through username and password or X.509 authentication transported at the headers of the SOAP messages through WSS4j API.

The Portal will prompt for authentication (login) for the protected resources, in this case, URLs invoking protected actions and private interfaces. The configuration of the protected URLs is done through a **security filter** that allows “form based like authentication” with a call to the Web service client in the background. The web service client caches the authenticated status for the duration of a session. When a Web session expires, authentication will be prompted again for the protected resources.

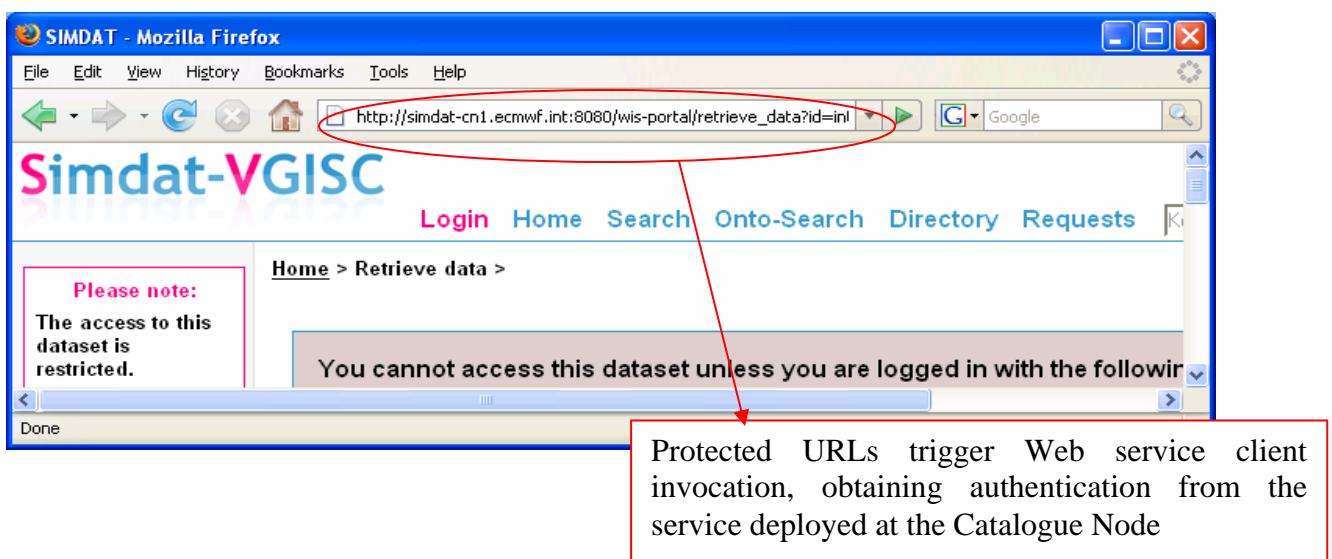


Figure 5-3: Protected Resources/URLs

---

## 5.4 Domain Authorities and Peer-to-Peer Secured Transmission.

The VMC messages are routed through several nodes following peer-to-peer algorithms. The security implementation takes into account this requirement, protecting messages with a digital signature that identifies the Node originating a request (endpoint authentication). A Domain Authority is a component that will sign a message before it is dispatched to another location. The Domain Authority generates SAML (Security Assertion Markup Language) expressions asserting that an authenticated user from a given node has certain roles or attributes. The assertion gets signed and ingested in the header of the request message by the Domain Authority in the originating Node. The intermediate peers would route the request message to the final destination where verification of the request message takes place. Before the Domain Authority signs the messages it is important that the XML messages are expressed in its canonical way to ensure verification of the message digest at the end point. The whole message then travels through an encrypted channel to ensure confidentiality.

## 5.5 Authorisation Factories

The VMC has implemented an Authorisation Factory model to support several authorisation schemes. The VMC supports several trusted domains. Within each domain, a particular implementation of an Authorisation scheme can be configured.

Three Authorisation schemes (*AuthorisationSolvers*) have been implemented so far:

- ***OpenSolver***: This authorisation scheme will check policies found in the metadata against those assigned to the user. No checks of domain membership are performed; therefore Nodes can live in an open space where no domain restrictions apply. The level of trust is “*open to everybody*”.
- ***RestrictedSAML Solver***: This Authorisation Solver will check domain membership of the Node originating the request. It will then extract the SAML assertion and validate the signature with the public key of the Node that generated the request (obtained initially at Domain formation). It will also verify the integrity of the message to ensure that it has not been tampered.

After validation and verification, it will extract the roles from the SAML assertion and intersect them with the roles expressed in the metadata record that describes the product requested. If a match is found, retrieval of the product requested will be granted.

- ***DACSolver***: The *DACSolver* provides authorisations based on a PDP (Policy Decision Point) component of the DAC framework developed by NEC. The PDP will receive an access Decision Request based on the user policies and Domain membership and grant access accordingly.

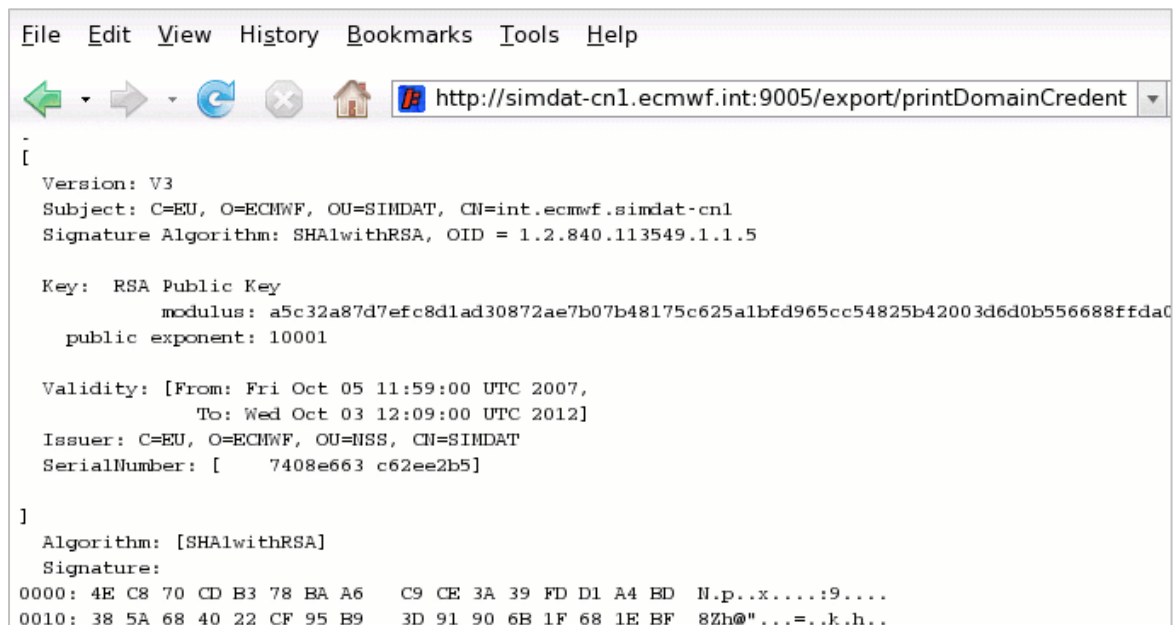
---

## 5.6 Management Tools

Tooling for the management of the VO was developed to ease the handling of users, domains and policies.

The management service offers two interfaces: a command line and a Web interface. This service is deployed at the Node component to facilitate administration tasks. The Web interface is protected by user name and password known only by administrators.

A facility to export and display the public key in the browser has been implemented. This facility will allow remote nodes to import public keys necessary to generate the domains and establish the trust.



**Figure 5-4 Printout of the public key. Remote Nodes can check the public key online before importing into their trusted domains.**

Command line utilities will allow importing the public key from a URL. The example below will add the *ecmwf* node to the *simdat\_demo* Domain, establishing a Trust relationship with the *ecmwf* node.

```
vmc-suite.sh node tools AddDomainMember
-domains simdat_demo
-site ecmwf
-url http://simdat-cn1.ecmwf.int:9005/export/getDomainPublicCertInBase64
```

---

## 6 Evaluation/Validation Report: Prototype at PM45.

Progress has been made and achievements have been reached in the reporting period inline with the requirements stated for Phase III of SIMDAT. The evaluation and validation done at PM36 summarised the pending list of requirements as follows:

### Grid Infrastructure

- Subscription Service
- Push mechanisms for critical data

### DDA

- Acquisition of real-time data
- Data Replication of critical data

### VO

- Assessment and consolidation
- Address cross-organisational issues
- Monitoring Tools

A review and summary check list validating the requirements that have been implemented since PM36 is provided.

### 6.1 Grid Infrastructure

The achievements in the Grid infrastructure layer have satisfied all the requirements stated for the last phase.

New services allowing the subscription of real time data have been deployed in the existing infrastructure. The push mechanism makes critical data available in a much shorter time frame than with the corresponding pull interface. The push interface introduces a notification framework where events are used to notify the data arrivals to the system and completeness of data deliveries to the end user.

#### 6.1.1 Requirements Validation

The new Grid infrastructure as of PM45 has been redeployed at the ECMWF Node in a test-bed that includes the Subscription Services and the new Data Repository interfacing with the GRIA based deployment.

The deployment has been used for testing the new capabilities of the Subscription Service (**Meteo Requirement P3.12 - Subscription Service**) and the GRIA Data Repository connection. Among the new subscription service capabilities, users are able to parameterize their request by subscribing to full/sub-selected datasets. After selecting the target dataset, users are able to choose a delivery method from his/her available destinations mechanisms. A request corresponding to the user's selection is placed and the dataset is delivered to the chosen destination as soon becomes available (**Meteo Requirement P3.3 - Push Mechanism for Critical Data**). Thanks to implementing a push data delivery method, data availability events are communicated internally and do not require user's intervention. A new version that includes the subscription is been tested and planned to be released by early summer.

The screenshot shows the Simdat-VGISC portal interface. At the top, there is a navigation bar with the logo 'Simdat-VGISC' and 'ECMWF'. Below the navigation bar, the user is logged in as 'vmc\_test'. The main content area is titled 'Home > Retrieve data >'. On the left, there are two informational boxes: 'This dataset is freely available.' and 'This dataset supports subscription.'. The main form is divided into several sections: 'Date' (with a date range from 1980-01-01 to 1990-12-31), 'Parameter' (with radio buttons for '2t' and 'msl'), 'Step time' (with radio buttons for '0000', '0500', '1000', and '1500'), and 'Base time' (with radio buttons for '0000', '0600', '1200', and '1800'). Below these sections are 'Shopping Cart' (with a checkbox for 'Deliver dataset to my shopping cart.'), 'FTP Delivery', and 'Email Delivery' (with a checkbox for 'My Email'). A 'Submit Request' button is located at the bottom of the form. Annotations with red boxes and arrows point to the 'Date', 'Parameter', 'Step time', and 'Base time' sections, and a list of two items: 'To choose Data Delivery Destination.' and 'To place the Subscription Request'.

Figure 6-1 New Subscription functionality integrated at the Portal.

Users can monitor (add, delete or modify) their subscription requests and manage (add and delete) their destinations (ftp, email, shopping cart options).

The screenshot shows the Simdat-VGISC portal interface. At the top, the user is logged in as 'vmc\_test'. The main content area is divided into sections for 'Requests' and 'DestinationsStatus'. The 'Requests' section shows two subscription requests, both with a green checkmark indicating they are completed. The first request is for 'Shopping Cart' and the second is for 'My Email'. The 'DestinationsStatus' section shows a table of destinations, including 'Shopping Cart' and 'My Email'. Red boxes and arrows highlight these elements, with callout boxes explaining their function.

**DestinationsStatus**

Name	Type	Login	Extra
Shopping Cart	ShoppingCart	vmc_test	
My Email	Email	vmc_test	email=Marta.Gutierrez@ecmwf.int name=VGISC user

Monitoring subscription requests (completed - *green button*).

Subscription Request delivering Data using the *Shopping Cart*.

Subscription Request delivering Data via *Email*.

Two Destinations: Shopping Cart and an Email address.

Figure 6-2 Managing Subscription Requests and Data Destination Mechanisms at the Portal.

By leveraging the SIMDAT infrastructure, access to GRIA based services have become available to any user worldwide. An authorized user can now submit and monitor requests, and retrieve the model results from the BRIDGE Server via the SIMDAT Portal (see next figure). The model outputs can be delivered in different standard image formats such as GIF, JPEG and KML.

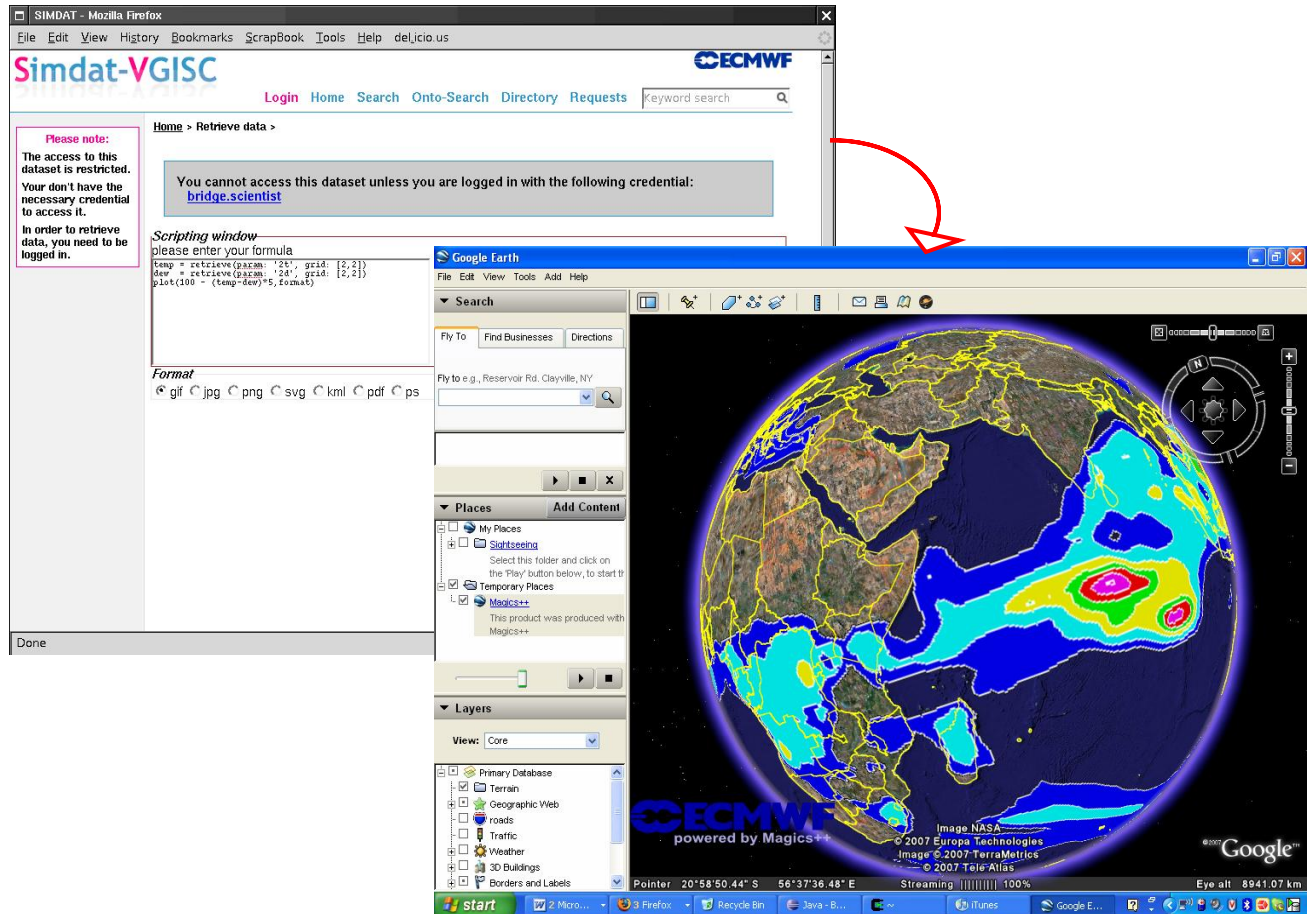


Figure 6-3 Accessing GRIA services via the SIMDAT Portal.

The left panel in the previous figure shows a “BRIDGE request” - an ensemble mean of precipitations computed in a distributed fashion- sent via the Portal to the BRIDGE Server. The right panel shows the outcome model, in KML format, visualized in Google Earth.

### Grid Infrastructure Achievements

The Grid infrastructure offers a personalized subscription service for delivery of critical data as soon as it becomes available.

The push mechanism implemented at the core of the Subscription Service’s data delivery uses the VMC communication channels more efficiently, thereby decreasing the traffic between Nodes.

The final Grid infrastructure provides full interoperability with diverse repositories including access to GRIA services that solve remote meteorological computations.

Grid Infrastructure requirements checklists:

Grid Infrastructure Requirements				
Requirement	Description	Priority	Implemented	Remaining
<b>Adopt Standard Interface</b>	Integration of web-service interfaces to ease interoperability.	High	✓	
<b>Interoperability Enhancement METEO-P3.1</b>	Provide access through the infrastructure to datasets served through OpeNDAP	High	✓	
<b>Backwards compatible software releases METEO-P3.2</b>	Provide compatibility with previous components	High	✓	
<b>Push mechanism for critical data. METEO-P3.3</b>	Implementation of push interfaces for weather warnings.	High	✓	
<b>Subscription service. METEO-P3.12.</b>	Implementation of a subscription service for delivering data as soon as they become available.	High	✓	
<b>GRIA Integration.</b>	Integration of GRIA-based services		✓	

## 6.2 Distributed Data Access

Multiple types of Data Repositories have been integrated with the SIMDAT Meteo Infrastructure. MeteoFrance has implemented a module for acquiring real time data from the GTS meteorological operational network (**Meteo Requirement P.3.15**). A metadata generation module allows converting WMO bulletins distributed by the MSSs into a data structure understandable by the VMC infrastructure. The generated metadata from the bulletins follows both the WMO-core profile and VGISC metadata schemas.

Replication and caching of critical data (**Meteo Requirement P.3.4**) is a requirement taken into account in the design of real-time and critical data acquisition module. This requirement will be addressed outside the SIMDAT project.

## 6.2.1 Requirements validation

A module that generates metadata from real time bulletins collected from the GTS (**Requirement METEO-P3.15**) has been implemented.

The GTS module collects the data from operational network to make this data available to the SIMDAT infrastructure. At the same time the same module has integrated a design for feeding data published at the SIMDAT infrastructure back into the GTS. Design for the replication/caching of some of this data has also been considered.

DDA Achievements
Data acquisition modules to publish real time datasets into the SIMDAT infrastructure have been implemented as well as metadata generation modules that extract the information from several WMO coded publications.

DDA Requirements				
Requirement	Description	Priority	Implemented	Remaining
<b>Data Replication/ Caching for Real time data</b>	Provide replication of real-time data to ensure high availability	High	Design provided	Out of scope
<b>Efficient transport of large amount of data</b>	Provide an study with protocols supporting large amounts of data transfers	High	✓	
<b>Synchronisation engine performance improvement</b>	Improve the synchronisation engine to provide better timings in the initialisation phase	High	✓	
<b>Acquisition of real time data</b>	To acquire real-time data from the GTS Message Switch Systems	High	Ongoing	Final Implementation for Prototype demonstration

### 6.3 Virtual Organisation and Ontology

The SIMDAT Meteo infrastructure deployed in Phase III has demonstrated the Virtual Organisation model. The VO concepts have been consolidated (**METEO-P3.6**) through the deployment at 8 sites. The VO version is part of the v0.9.8.1 distribution (see ANNEX-I: Software Releases), at the same time the previous version has been running in a parallel deployment.

Monitoring requirements **METEO-P3.9.a** will be addressed outside the SIMDAT project. Although the software has an identifiable set of functions that would report into a monitoring module, this interface will be addressed outside SIMDAT.

	<b>v0.9.7</b>	<b>V0.9.8 + patch (VO)</b>
<b>ECMWF (UK)</b>	Y	Y
<b>DWD (Germany)</b>	Y	Y
<b>EUMETSAT (Germany)</b>	Y	Y
<b>MetOffice (UK)</b>	Y	Y
<b>MeteoFrance</b>	Y	Y
<b>JMA (Japan)</b>	Y	Y
<b>CMA (China)</b>	Y	X
<b>BOM (Australia)</b>	Y	X
<b>KMA (korea)</b>	Y	X
<b>NCAR (US)</b>	<b>OAI Harvesting</b>	<b>OAI Harvesting</b>
<b>RNODC (Russia)</b>	<b>OAI Harvesting</b>	<b>OAI Harvesting</b>
<b>LPMG (Portugal)</b>	X	Y
<b>NEC (Germany)</b>	X	Y

Table 6-2: VO Prototype deployment -v0.9.8

VO and Ontology Requirements				
Requirement	Description	Priority	Implemented	Remaining
<b>VO Assessment and consolidation METEO-P3.6</b>	Deploy the VO infrastructure at all partners	High	90%	Deployment at 8 sites
<b>Metadata Integration of the data access policy. METEO-P3.7</b>	Integrate data access policies in the metadata records	High	✓	
<b>Development of tools for user and policy administration. METEO-P3.8</b>	Develop administration tools to register users and policies within the Domains.	High	✓	
<b>Address cross-organisational issues METEO-P3.9</b>	Provide support for cross domain authentication and authorisation		✓	Full testing environment to be done with several domains at several organisations.
<b>Monitoring Tools METEO-P3.9a</b>	Develop tools to monitor the synchronization, data delivery and VO services	High	Ongoing	Out of scope
<b>Ontology based catalogue navigation METEO-P3.10a</b>	Integration of the ontology in the Portal	High	✓	
<b>Discovery Service enhancement METEO-P3.10b</b>	Text Search improved to provide better results.	High	✓	
<b>Thesaurus Maintenance Service METEO-P.3.14</b>	Implementation of a service to manage Ontologies	High	✓	
<b>Metadata Editor METEO-P3.11</b>	Implement web interface to generate metadata records compliant to the WMO.	High	✓	

---

Ontology/Discovery and VO Achievements
Thesaurus and Ontology navigation with multilingual functionality has been integrated in the Portal providing better navigation through meteorological concepts.
The deployment of the VO has been performed at 9 sites, establishing several Trust Domains that validate the security model restricting access according to policies and Domain membership in a fully distributed environment.
An standard search interface based on ISO23950 has been implemented to interoperate with the GEOOS community and other WMO programs.

## 7 Final Prototype

As reported in this document the SIMDAT Meteo activity has finalised most of the requirements in terms of developments. The latest achievements are being tested in a test-bed based at ECMWF.

A final version including the subscription service will be released, so the partners and collaborators can deploy the new infrastructure. The final prototype will be inline with the demonstrator activities. Installation of this version will require integration with the legacy systems and configuration with a legacy data delivery system. Extensive documentation and support will be given to the partners to be able to finalise deployment activities and integration with the new version.

The remaining activities planned until the end of the project will focus on the documentation of an extensive guide for developers and administrators, and on the packaging of the software. The documentation will be a good guideline to support the final prototype and smooth deployment and integration.

## 8 Conclusion

SIMDAT Meteo has reached its final stage by introducing the new subscription service. Based on the same communication services and message exchange platform, the subscription service has leveraged the existing distributed infrastructure and extended most of the components to allow for timely delivery of real time datasets.

The resulting software has proven to be an extensible framework to deploy services in a well integrated and configurable platform. Therefore, all the services deployed in the Virtual Meteorological Centre during the life time of the project have always been based on the same well integrated software stack. The extensibility of this framework has allowed to “plug and display” a wide array of heterogeneous data repositories, exposed to SIMDAT through a variety of service interfaces such as GRIA based meteorological services at ECMWF, OGSA-DAI at DWD, WMS (Web Map Server) at the UKMO, OPENDAP at KMA just to name a few.

Thanks to this service oriented platform and architecture all the distributed aspects and complexities of the underlying system are completely transparent to the user. The end user instead has gained access to a much wider set of meteorological data that was beforehand sitting at the individual data centres.

---

In addition to the Subscription service, services like Ontology, Trust Domains, Discovery, Peer-to-Peer routing, Synchronisation, Authentication, Authorisation, Asynchronous Data Retrievals ... have also made possible the transparency to the end user in such a complex environment.

## 9 REFERENCES

[1] **D.12.3.1** Document detailing the requirements, providing a first design and outlining the architecture of the Subscription Service. It also outlines the major modifications to the VMC implementation previous to the subscription service implementation.

[2] **D.20.1.3** Document providing a technical outline of the design of an alpha version of the meteorological demonstrator.

[3] **D.12.2.5** Document listing the achievements and lessons learned from the prototype PM30. It gives details on the validation of the main components of the PM30 prototype and defines the scenario for SIMDAT phase III.

[4] **D12.2.2** Document describing the model approach adopted to describe the VO by the Meteorological partners. The document also presents architectural view of the VO security services requirements.

[5] **D.12.2.3** Document that formalises the VO security requirements for the first phase of the project. The document also describes the technical solution and prototype implementation that will address these requirements.

## 10 Glossary

<b>Term</b>	<b>Definition</b>
<b>AODV</b>	Ad hoc On-Demand Distance Vector Routing
<b>AHL</b>	Abbreviated Heading Line
<b>AMS</b>	American Meteorology Society
<b>CN</b>	Catalogue Node
<b>CQL</b>	Common Query Language
<b>DDA</b>	Distributed Data Access
<b>DR</b>	Data Repository
<b>GCMD</b>	Global Change Master Directory
<b>GEMET</b>	General Multilingual Environmental Thesaurus
<b>GEOOS</b>	Global Earth Observing System of Systems
<b>GISC</b>	Global Information System Centre
<b>GRIA</b>	Grid Resources for Industrial Applications.
<b>GTS</b>	Global Telecommunication System
<b>MSS</b>	Message Switching System
<b>OPENDAP</b>	Open-source Project for a Network Data Access Protocol
<b>PDP</b>	Policy Decision Point
<b>PKI</b>	Public key Infrastructure
<b>SAML</b>	Security Assertion Markup Language
<b>SECURITY FILTER</b>	<a href="http://securityfilter.sourceforge.net/">http://securityfilter.sourceforge.net/</a>
<b>SKOS</b>	Simple Knowledge Organisation System
<b>SOAP</b>	Simple Object Access Protocol
<b>STS</b>	Security Token Service
<b>SWEET</b>	Semantic Web for Earth and Environmental Terminology
<b>UNIDATA</b>	<a href="http://unidata.ucar.edu">http://unidata.ucar.edu</a>
<b>VGISC</b>	Virtual Global Information System Centre
<b>VMC</b>	Virtual Meteorological Centre
<b>Volume C1</b>	WMO N°.9 <i>Volume C1</i> Reference Catalogue
<b>WIS</b>	WMO Information System
<b>WMO</b>	World Meteorological Organisation
<b>WSS4j</b>	An implementation of the OASIS Web Services Security (WS-Security)
<b>X509</b>	ITU-T standard for a public key infrastructure (PKI) and Privilege Management Infrastructure (PMI)

## 11 ANNEX I: Software releases

Version	Release Date	Major Changes
Release 1.0	July 08	This version release includes: <ul style="list-style-type: none"> <li>▪ The interface for the new subscription services.</li> <li>▪ Upgrade of the VGISC metadata extension incorporating the subscription elements.</li> </ul>
Patch/release v0.9.8.1.b	July 08	This version and patch release includes: <ul style="list-style-type: none"> <li>▪ New rendering widgets.</li> <li>▪ Fixes of several bugs.</li> <li>▪ Error control.</li> </ul>
Patch v0.9.8.1a	November 07	This version patch includes: <ul style="list-style-type: none"> <li>▪ Two management tools for registering users: A Web Interface and a Command Line interface.</li> <li>▪ “Man pages” for all start up scripting and management tools.</li> </ul>
Patching System Release	November 07	This version release includes a patching system that makes easy upgrading from one version to the next.
0.9.8.1	October 07	<ul style="list-style-type: none"> <li>▪ Integration of security components for supporting the VO model.</li> <li>▪ Integration of Thesaurus and Ontologies to augment the existing search facility at the Portal.</li> </ul>
0.9.7	February 07	<ul style="list-style-type: none"> <li>▪ Support for dynamic path discovery on the mesh network via integration of the AODV.</li> <li>▪ Support for filtering functions in the synchronisation engine (e.g. filter metadata by owner).</li> <li>▪ Improved XML parsing technique.</li> </ul>
v. July 06	July 2006	<ul style="list-style-type: none"> <li>▪ Major enhancements of the node providing a stable synchronisation engine based on a journalised system and with support for a mesh network allowing communication between nodes on a Peer to Peer basis.</li> <li>▪ A zero development data repository where integration of existing data access is reduced to the minimum development for quick deployment and set-up.</li> <li>▪ An internet portal offering a discovery and data access service.</li> <li>▪ Support for the ingestion of several metadata formats (THREDDS, Dublin Core, WMO Core Profile version 0.2 and 0.3 and E2EDM )</li> </ul>

---

<b>Version</b>	<b>Release Date</b>	<b>Major Changes</b>
		through configuration files that extract discoverable information through XPATH expressions.
Initial Version		<ul style="list-style-type: none"><li>▪ The first release of the software included a data repository and catalogue node with a first version of the synchronisation engine.</li></ul>