

Proposal/Contract: 031874

CYCLOPS

CYber-Infrastructure for Civil protection Operative Procedures

Dissemination and coordination activity final report

(Deliverable D17)

Reference : CYCLOPS-WP02-D17-INFN-002

Due date of deliverable: 30 / 09 / 2008

Actual submission date: 15 / 11 / 2008

Start date of the project: **JUNE 1st 2006** Duration: **30 months**
 Partner: **INFN**
 Editor(s): **Marco Verlato**
Issue: 01 Revision: 00

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



REF:CYCLOPS-WP02-D17-INFN-002

Issue: 01 Rev: 00

Date: 24/11/2008

PAGE INTENTIONALLY LEFT BLANK



REF:CYCLOPS-WP02-D17-INFN-002

Issue: 01 Rev: 00

Date: 24/11/2008

DOCUMENT HISTORY

Issue.revision	Date	Description of change	Author	Affiliation
0.1	01 Sep 2008	First draft	Marco Verlatto	INFN
0.2	8 Sep 2008	Annex I added	Stefano Dal Pra, Valerio Angelini, Mirco D'Andrea	INFN, CNR- IMAA, CIMA Foundation
0.3	12 Sep 2008	Annex II added	Stefano Dal Pra, Eric Frizziero, Silvano Squizzato	INFN
0.4	9 Oct 2008	Overall review	Valerio Angelini	CNR-IMAA
0.5	30 Oct 2008	University of Minho contribution added	Antonio Pina, António Esteves, Vítor Sá, Ricardo Marques	UMINHO
0.6	11 Nov 2008	Grid enabled Alhtair contribution added	Vincent Thierion, Pierre-Alain Ayrat	EMA-Armines
0.7	12 Nov 2008	Overall review	Marco Verlatto	INFN
1.0	24 Nov 2008	Review	Project Review Board	all



TABLES OF CONTENTS

1	INTRODUCTION	6
1.1	PURPOSE OF THE DOCUMENT.....	6
1.2	APPLICATION AREA	6
1.3	REFERENCES.....	6
2	EXECUTIVE SUMMARY	7
3	DISSEMINATING EGEE TOWARDS CIVIL PROTECTION COMMUNITY.....	8
3.1	CYCLOPS OPEN CONFERENCE.....	8
3.2	CYCLOPS FIRST TRAINING WORKSHOP	8
3.3	INFN TRAINING COURSE ON PRACTICAL ASPECTS OF APPLICATION'S INTEGRATION IN GRID	9
3.4	CYCLOPS SECOND TRAINING WORKSHOP	10
4	COORDINATION BETWEEN CYCLOPS AND EGEE ACTIVITIES.....	12
4.1	DISSEMINATION OF CYCLOPS TOWARDS GRID COMMUNITY	12
4.2	THE CYCLOPS VIRTUAL ORGANISATION	13
4.3	CONTACTS WITH OTHER GRID PROJECTS	14
5	CONCLUSIONS.....	16
6	ANNEX I: REPORT OF INFN TRAINING COURSE ON PRACTICAL ASPECTS OF APPLICATION'S INTEGRATION IN GRID.....	17
6.1	APPLICATION IDENTITY CARD	17
6.2	APPLICATION GENERAL DESCRIPTION	17
6.3	PORTING WORK DESCRIPTION.....	18
6.3.1	Goal	18
6.3.2	Grid porting strategy	18
6.3.3	Description.....	18
6.3.4	gJobSubmitter Module.....	20
6.3.5	Final considerations on the adopted submission strategy	20
6.4	PERSONAL COMMENTS	21
7	ANNEX II: DOCUMENT DESCRIBING THE GRID LAYER PROTOTYPE DEVELOPED TO REALIZE THE CYCLOPS DEMONSTRATION.....	22
7.1	TERMINOLOGY	22
7.2	WORKFLOW OUTLINE	22
7.2.1	Involved components.....	22
7.3	THE WRAPPER	23
7.3.1	Builtin capabilities	24
7.3.2	Configuring wrapper with JSON	24
7.3.3	Motivation.....	24
7.4	LAYER TO ACCESS TO GRID INFRASTRUCTURE	25



7.5	JOBMANAGEMENTTOOL CLASS.....	26
7.5.1	JobManagementTool API.....	26
7.6	JOBSUBMISSIONHELPER CLASS.....	28
7.6.1	JobSubmissionHelper API.....	28
7.7	CONFIGURATIONFILE CLASS.....	29
7.7.1	ConfigurationFile API.....	29
7.7.2	The CYCLOPS' configuration file.....	29
7.8	HOW TO USE THE GRID LAYER.....	30
7.8.1	Instanting a JobManagementTool object.....	30
7.8.2	Creating the job from JDL.....	30
7.8.3	Submitting the job.....	30
7.8.4	Retrieving the output.....	31
7.9	SOFTWARE DOWNLOAD.....	32
7.10	BIBLIOGRAPHY.....	33
8	ANNEX III: UNIVERSITY OF MINHO CONTRIBUTION.....	34
8.1	GRID INFRASTRUCTURE.....	34
8.1.1	EGEE ROLL.....	34
8.1.2	CP Virtual Organization.....	35
8.1.3	Conclusions.....	36
8.2	INTEGRATION OF SENSORS.....	37
8.2.1	Approach.....	37
8.2.2	System Architecture.....	38
8.2.3	System Prototype.....	39
8.2.4	Case Studies.....	40
8.2.5	Conclusions.....	42
8.3	PARTNERSHIP: ÉCOLE DE MINES – UNIVERSITY OF MINHO.....	42
8.3.1	Flash Flood.....	42
8.3.2	Alhtair.....	44
8.3.3	A proposal for the parallelization of Ahltair.....	44
8.3.4	Flood forecasting process through Grid Enabled OWS.....	45
8.3.5	Conclusions.....	47
8.4	COLLABORATION DORII – UNIVERSITY OF MINHO.....	47
8.4.1	Problems addressed and results.....	47
8.5	COLLABORATION NGI / EELA-2 / INT.EU.GRID – UNIVERSITY OF MINHO.....	48
8.5.1	Portuguese NGI.....	49
8.5.2	Int.EU.Grid.....	50
8.5.3	EELA-2.....	50

1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

According to the Description of Work (DoW) of the CYCLOPS project [R1], pag. 43:

“this deliverable is a document reporting the results of the dissemination of EGEE towards the Civil Protection community, and about the coordination between the EGEE and CYCLOPS activities”

The document was due as project deliverable D17 [R2], and was under the responsibility of the WP2 leader partner, INFN.

1.2 APPLICATION AREA

The main target audience of this document are the CYCLOPS project officer and the CYCLOPS partners, but all stakeholders in the CYCLOPS project are potential recipients of EGEE dissemination and training events.

1.3 REFERENCES

[R1]	http://www.cyclops-project.eu/Default.aspx?id_menu=9&id_documento=35&id_versione=1	CYCLOPS Annex I
[R2]	http://www.cyclops-project.eu/	CYCLOPS Project home page
[R3]	http://www.cyclops-project.eu/Default.aspx?id_menu=9&id_documento=28&id_versione=1	D4 – Dissemination Plan
[R4]	http://www.cyclops-project.eu/Default.aspx?id_menu=9&id_documento=29&id_versione=1	D5 – Training Events Plan
[R5]	http://www.cyclops-project.eu/Default.aspx?id_menu=9&id_documento=30&id_versione=1	D7 – EGEE Cookbook
[R6]	http://www.cyclops-project.eu/Default.aspx?id_menu=9&id_documento=26&id_versione=1	D2 – Report on Open Conference
[R7]	http://grid.infn.it/	INFN-Grid web site
[R8]	http://www.cyclops-project.eu/Default.aspx?id_menu=9&id_documento=33&id_versione=1	D10 – 1 st Training Workshop Report
[R9]	https://gilda.ct.infn.it	Grid INFN Laboratory for Dissemination Activities
[R10]	http://www.cyclops-project.eu/Default.aspx?id_menu=9&id_documento=44&id_versione=1	D12 - Mid-term Project Workshop Report
[R11]	http://www.cyclops-project.eu/Default.aspx?id_menu=9&id_documento=43&id_versione=1	D13 – 2 nd Training Workshop Report
[R12]	To appear in http://www.cyclops-project.eu/Default.aspx?id_menu=9	D14 - EGEE Request for enhancement document
[R13]	http://gridice.forge.cnaf.infn.it/	GridICE web site
[R14]	http://www.cyclops-project.eu/Default.aspx?id_menu=9&id_documento=32&id_versione=1	D9 – Use Cases document
[R15]	http://www.cyclops-project.eu/Default.aspx?id_menu=9&id_documento=34&id_versione=1	D11 – System Requirements document

2 EXECUTIVE SUMMARY

This deliverable reports about the dissemination activity carried out by CYCLOPS project, and the action taken to assure coordination between CYCLOPS and EGEE activities.

The adopted methodology to carry out the dissemination activity, together with its goals, sources, targets, content and media have been already described in the deliverables D4 [R3] and D5 [R4].

The rest of the document is organised as follow:

Section 3 is about the dissemination of EGEE results towards Civil Protection community.

Section 4 describes the coordination of CYCLOPS project with EGEE activities.

Section 5 draws the conclusions.

Section 6 is an annex reporting the experience done at the “INFN Training Course on practical aspects of application’s integration on grid”

Section 7 is an annex document detailing the grid layer prototype developed for the CYCLOPS demonstration carried out at IBERGRID’08, EGEE’08 and SC08 Conferences.

Section 8 is an annex describing the contribution of University of Minho partner to CYCLOPS activities (including their collaboration with EMA-Armines on Flash Flood use-case) after they joined the project in December 2007.

3 DISSEMINATING EGEE TOWARDS CIVIL PROTECTION COMMUNITY

This activity was planned in the DoW [R1] in order to make GMES and Civil Protection communities aware of the EGEE project and its distributed grid computing infrastructure.

The current EGEE platform is based on the middleware developed by several pioneering projects, most notably DataGrid, where the main focus was put on enabling to grid computing the data intensive scientific applications in the fields of High Energy Physics, Earth Observations and Genomic Exploration. Their requirements lead to a platform designed to handle the processing of a large amount of distributed data, ranging in size from 0.1 to 10 PetaBytes/year, exploiting the sharing of computing, storage and network resources in a coordinated way.

This dissemination activity, carried out as task T2.1 of Work Package 2, was intended to provide an example of grid platform to evaluate and test its capabilities in the GMES context considering the Civil Protection applications, describing in detail the current architecture of the EGEE grid middleware and its planned evolution.

As a result of this activity, a document describing the EGEE platform and guiding Civil Protection grid users to move their first steps into EGEE grid was delivered at project month 6 as deliverable D7: “EGEE Cookbook: a guide for Civil Protection grid users” [R5].

Dissemination of EGEE towards GMES and Civil Protection communities throughout the lifetime of the CYCLOPS project took place at the following four events:

- CYCLOPS Open Conference, Rome (Italy), 16/6/2006
- CYCLOPS First Training Workshop, Bologna (Italy), 11-13/4/2007
- I Corso di formazione INFN su aspetti pratici dell'integrazione di applicazioni in grid (INFN Training Course on practical aspects of application's integration in grid), Martina Franca (Italy), 12-23/11/2007
- CYCLOPS Second Training Workshop, Chania (Greece), 5-7/5/2008

A summary of the above events is given below.

3.1 CYCLOPS OPEN CONFERENCE

The Project Open Conference was held in Rome the 16/6/2006, two weeks after the official start of CYCLOPS. It was intended to present CYCLOPS to GMES and Civil Protection communities. A full report is available as deliverable D2 [R6]. During the conference, Prof. Mazzucato, member of the EGEE management board and director of INFN-Grid [R7], presented the EGEE project. He introduced the EGEE computing grid which operated and included at that time more than 200 sites in 39 countries. This makes it the world's largest international scientific grid and represents an important and strategic sector for GMES, and in particular Civil Protection operative infrastructures. He presented related EU projects, EGEE applications and the outreach activities. The full presentation is available at: <http://www.cyclops-project.eu/demo/pdf/EGEE.pdf>

3.2 CYCLOPS FIRST TRAINING WORKSHOP

The First Training Workshop was held in Bologna (Italy) the 11-13/4/2007, at the conference room of the INFN-CNAF centre. A full report is available as project deliverable D10 [R8].

The first half day of the workshop was dedicated to a number of presentations focused on the CYCLOPS activities, and given by the Project Manager and the CNR-IMAA, DDSC-EMA and DPC representatives.

The training course proceeded then touching in deep the four main areas of the gLite middleware architecture: Security, Information System, Data Management and Job Management. Theoretical lessons were alternated with practical exercises experienced by each student from his own laptop connected to the GILDA infrastructure via the wireless network link made available by INFN-CNAF structure hosting the event.

GILDA (Grid INFN Laboratory for Dissemination Activities) [R9] is a training infrastructure (t-infrastructure) spread over three continents and since the beginning of EGEE a reference point for dissemination and training activities. This infrastructure was made available from INFN as main instruments for giving the tutorial. A t-infrastructure, i.e. an e-

infrastructure for knowledge dissemination, is a complete Grid environment with a set of special features and specific solutions for knowledge dissemination available to teachers and students with devoted solutions. Students logged in their pre-defined accounts into the GILDA User Interface hosted at INFN-Catania, owned a X509 personal certificate issued by the fake GILDA Certification Authority and made the exercises on the GILDA test-bed exploiting the GILDA Virtual Organisation. The lectures and the exercises were presented by the INFN GILDA team composed by Dr. Emidio Giorgio (INFN-Catania), Dr. Tony Calanducci (INFN-Catania), Dr. Fabio Scibilia (INFN-Catania) and Dr. Enrico Fattibene (INFN-CNAF).

A special session dedicated to newly developed tools to enable Relational Database access via grid (a major feature foreseen to become available in the next releases of the gLite middleware) was scheduled in the second day of the workshop. Dr. Sandro Fiore of University of Lecce presented the GRelC middleware, while Dr. Claudio Vuerli of INAF-Trieste presented the G-DSE middleware. Both products adopt the same security mechanism of gLite and are capable of publishing their status on the gLite information system, making then easy their integration in a gLite based infrastructure.

The final half day was dedicated to illustrate the APIs used to enable the porting of the application software over gLite. Some success stories and examples of application in different scientific fields already ported in gLite were also presented.

A total of 14 people attended the event, 8 of them coming from institutions related with the CYCLOPS activity. The participation was in fact open also to interested people not strictly related to the CYCLOPS project.

Being the main target of the event the representatives and ICT developers of the CYCLOPS partner institutions, the participation was unfortunately less than expected. It was mainly due to financial administrative issues which prevented the CYCLOPS partners from Portugal and Greece (ANPC, CP-CH and TEI-CR, three out of the seven initial CYCLOPS partners) to attend the event.

At the end of the course the participants were asked to fill an anonymous feedback form in order to assess the entire training event and provide comments and suggestions for further training events. This form allowed to rate each one of the lessons presented at the course, and to evaluate Event Materials, Lab Exercises, Advertising & Registration, Facilities and the entire event. For each of the above items and edit box is available for writing comments, and a score ranging for 1 to 6 has to be assigned (1: Poor, 2: Insufficient, 3: Fair, 4: Good, 5: Very Good, 6: Excellent). The overall evaluation parameter resulted in a score of 5.

The link with the complete agenda and all of the presentation in PPT and/or PDF format is available from the project portal www.cyclops-project.eu → left side menu “Events” → “First Training Workshop”

3.3 INFN TRAINING COURSE ON PRACTICAL ASPECTS OF APPLICATION'S INTEGRATION IN GRID

The INFN Training Course tailored for application porting on grid was held in Martina Franca (Italy) the 11-23/11/2007. This two weeks school represented a great opportunity to learn about grid architecture and to practice with its available instruments in order to test what useful potential it offers for Civil Protection issues.

It was attended by Dr. Mirko D'Andrea of CIMA Foundation, a research institution collaborating with Italian DPC, Dr. Valerio Angelini of CNR-IMAA and Dr. Stefano Dal Pra of INFN-Padova.

At the end of the school, an application in use with the Italian Civil Protection called RISICO was ported to run in the GILDA grid environment.

RISICO is a prevision system designed for the assessment of the distribution of dynamic wild land fire risk over the whole Italian territory which has been developed by the CIMA Foundation for the Italian Civil Protection Agency. Such an assessment takes place on the basis of static information concerning the vegetation cover and the topography of the territory and of dynamic information consisting in the meteorological forecast, over a certain time horizon (72 hours), provided by a Limited Area Model. RISICO performs his prediction about potential dangerousness over a discretized domain of 1 km cells (these are about 330.000) covering the whole Italian territory. The model is intrinsically parallelizable being that computation only takes into account cell status without considering correlations between them. Input data for the model are provided by a network of meteo sensors as well as atmospheric models (Local Area

Model). These data gets daily retrieved and aggregated. RISICO also needs a static set of information describing cells respect to orographic and vegetation characteristics for the discretized domain. The adopted resolution for cells influences accuracy for static variables represented by the territory portion covered by cells. This has a strong impact especially for vegetation, being that Mediterranean regions have a great variety of different coexisting species as also a great importance in determining the fire risk estimation. RISICO provides its computation output as a set of maps representing predicted behavior in space and time for those variables which are useful in identifying risky situations. At present these variables indicates potential fire risk, i.e. the danger level of a fire, without considering probability of it to begin. Output data produced for the whole Italian territory takes about a daily amount of 100MB, with an overall 40MB of needed input data, when using 1km cells. Needed computation time is in the order of 40 minutes using a common workstation.

The benefit of using grid to run RISICO consisted in achieving a significant improvement in the accuracy (reducing cell size from 1km to 100m) keeping the same or less total computation time.

A full report describing the porting strategy and the results achieved during the course is given as annex to the present document in section 6.

The work carried out during this school was the basis of further developments bringing to G-RISICO, the grid-enabled version of RISICO leveraging on Open Geospatial Consortium (OGC) Web Services presented at the Second Training Workshop and then at the joint IBERGRID'08 / Mid-term CYCLOPS Project Conference held in Porto the 12-14/5/2008 [R10].

3.4 CYCLOPS SECOND TRAINING WORKSHOP

The Second Training Workshop was held in Chania (Greece) the 5-7/5/2008, at the conference room of the Panorama Hotel. A full report is available as project deliverable D13 [R11].

The first half day of the workshop was dedicated to a number of presentations focused on the CYCLOPS activities, and given by the local hosting CYCLOPS partner TEI-Crete, the Project Manager and the CNR-IMAA, EMA and CIMA-DPC representatives. A talk from a DORII project representative was also scheduled, being the activity of DORII well related with Civil Protection applications.

The opening of the workshop was made by two welcome speeches from the Vice Rector of TEI-Crete Institute, Prof. Kaliakatsos, and from the Head of Civil Protection of Chania Prefecture, Mr. Koukianakis.

The training courses proceeded then touching in deep the four main areas of the gLite middleware architecture: Security, Information System, Job Management and Data Management. Theoretical lessons were alternated with practical exercises experienced by each student from his own laptop or desktop workstation connected to the GILDA infrastructure via cable or wireless network link made available by the local structure hosting the event. A session describing the newly developed Secure Storage Service was also scheduled. This new service integrated with the gLite middleware has been designed to solve the insider problem on data saved in remote hosts, in order to guarantee confidentiality and integrity of data. In fact the service permits confidential data to be stored in an encrypted format in a Storage Element and accessed only by its owners or authorised users. This capability emerged as request for enhancement of the current EGEE platform to satisfy Civil Protection requirements, as analysed and described in the project deliverable D14 [R12]. The lectures and the exercises were presented by the INFN team composed by Dr. Tony Calanducci, Dr. Elisa Ingra and Dr. Stefano Dal Pra.

The layout of the course was very similar to the one of the First Training Workshop. It was in fact mainly targeted to people mobilised from those CYCLOPS partners (ANPC, CP-CH and TEI-CR institutions) that could not attend the previous event, and to the new partner (University of Minho) which joined in the second year of the project.

A dedicated session to provide guidelines for joining the real production grid infrastructure and in particular the official recognized CYCLOPS Virtual Organisation, both as users and site managers, was also scheduled and presented by Dr. Marco Verlato of INFN.

The final half day has seen the presentations of the application porting strategies carried out to grid-enable to two CYCLOPS selected use cases, the SPC-GD and RISICO application. Being the last in a more mature level, a live demonstration of the application running on the grid was also given to the audience. It was an useful rehearsal for a further live demo scheduled at the IBERGRID'08 Conference in the following week.

A total of 31 people attended the event, most of them coming from the Technological Institute of Crete. The overall evaluation parameter, computed from anonymous feedback forms as described in section 3.2, resulted in a score of 4.45.

The link with the complete agenda and all of the presentation in PPT and/or PDF format is available from the project portal www.cyclops-project.eu → left side menu “Events” → “Second Training Workshop”



4 COORDINATION BETWEEN CYCLOPS AND EGEE ACTIVITIES

A fundamental aspect of GMES and grid community inter-dissemination is the knowledge exchange with EGEE, the main grid project at European level. CYCLOPS members developing the Civil Protection grid framework must not only be kept continuously updated about new middleware functionalities implemented by EGEE project, but need also to give feedback to EGEE developers highlighting the weakness encountered in the grid middleware and suggesting possible improvements for satisfying their specific requirements. To achieve that, several reports about CYCLOPS activities and results were presented to the grid community throughout the lifetime of the project, actively participating to the most important international grid related conferences and workshops.

The first opportunity to illustrate the CYCLOPS roadmap to the EGEE leader Bob Jones was however the Technical Expert Group meeting organised at DPC quarters in Rome the 30th of November 2006. The outcome of the meeting was a precious guidance for the following of the project. The main feedback from grid community on the proposed roadmap was in fact summarized in the suggestions listed below:

- to work with EGEE people to be in contact with other grid projects which have different objectives but share common issues with CYCLOPS
- to select and further investigate/prototype a couple out of the five use cases presented at the Open Conference, focusing in particular on Flash Flood and Forest Fires
- to start from the present gLite middleware to study the implementation of Open Geospatial Consortium Web Services (OWS) on the top of a grid platform, suggesting inputs for the future EGEE platform
- to keep Geosciences and GMES communities informed about CYCLOPS activities

4.1 DISSEMINATION OF CYCLOPS TOWARDS GRID COMMUNITY

Here below is the complete list of grid related events where oral and/or poster presentations about CYCLOPS were accepted:

- EGEE'06 Conference, 25-29/9/2006, Geneva - Switzerland
- Jornada InGrid '06, 11/11/2006, Braga - Portugal
- 2nd GRID & e-Collaboration Workshop for the Earth Science and the Space Community, 11-12/12/2006, Frascati - Italy
- Open Grid Forum 20, 7-11/5/2007, Manchester - UK
- 2nd EGEE User Forum, 9-11/5/2007, Manchester - UK
- EGEE'07 Conference, 1-5/10/2007, Budapest – Hungary
- SC07: International Conference on High Performance Computing, Networking, Storage and Analysis, 10-16/11/2007, Reno - USA
- 3rd GRID & e-Collaboration Workshop for the Earth Science and the Space Community, 16-17/1/2008, Frascati - Italy
- 3rd EGEE User Forum, 11-14/2/2008, Clermont-Ferrand - France
- IBERGRID'08: 2nd Iberian Grid Infrastructure Conference, 12-14/5/2008, Porto - Portugal
- Open Grid Forum 23, 2-6/6/2008, Barcelona - Spain
- EGEE'08 Conference, 22-26/9/2008, Istanbul - Turkey

- SC08: International Conference on High Performance Computing, Networking, Storage and Analysis, 15-21/11/2008, Austin - USA

The live demonstration presented at EGEE'08 Conference, "*G-RISICO: A Wild Fire Risk Assessment application running on an advanced Grid Infrastructure*", was awarded with the "Best Demo Prize".

A complete list with the web links to the above events and the material presented in PPT and/or PDF format is available from the project portal www.cyclops-project.eu → left side menu "Events".

According to the suggestions of the Technical Expert Group, CYCLOPS dissemination activity was carried out also towards not strictly grid related communities, e.g. giving presentations at the European GeoInformatics Workshop (2007), at the XXIV General Assembly of the International Union of Geodesy and Geophysics (IUGG 2007), at the General Assembly of the European Geosciences Union (EGU 2008).

The full list of events where CYCLOPS was presented is available on the project portal.

4.2 THE CYCLOPS VIRTUAL ORGANISATION

Another aspect of the coordination activity with EGEE has been the deployment and support of the CYCLOPS Virtual Organisation (VO) and its operations on the EGEE production grid infrastructure. Some high-level gLite services of the Italian production INFN-Grid infrastructure [R7] (one of the Regional Operation Centre of EGEE grid) have been in fact enabled with the CYCLOPS VO, these are:

- WMS hosted by prod-wms-01.pd.infn.it
- LB hosted by prod-lb-01.pd.infn.it
- BDII hosted by prod-bdii-01.pd.infn.it
- VOMS server (<https://voms2.cnaf.infn.it:8443/voms/cyclops/>)
- LFC catalogue hosted by lfcserver.cnaf.infn.it

These high level services allow for jobs submission and data management over a number of computing and storage resources made available by nine Italian, two Portuguese and one French grid sites hosted by institutions involved in the CYCLOPS project, which decided to support the CYCLOPS VO and to share their resources.

A guide for CYCLOPS VO users, describing in detail the whole infrastructure and services, and a guide for helping site managers to deploy gLite resources enabling CYCLOPS VO have been written and published as wiki pages.

The current status of the grid sites supporting the CYCLOPS VO is continuously monitored by the GridICE tool [R13], which shows updated information about queues occupancy, storage capacity and job details. Concerning the job accounting, the picture below shows the cumulative total number of jobs run at the various sites as a function of time from October 2007 to September 2008. The figure 4-1 is taken from the official EGEE Accounting System, demonstrating that CYCLOPS VO is well integrated into the EGEE grid infrastructure and its operations. More than 7000 jobs for a total of about 50 CPU.days were run in the considered period, and the deep rise in the number of jobs between April and May 2008 is due to the preparation and testing of the G-RISICO demonstration carried out in May at the Second Training Workshop and then at IBERGRID'08.

A description of the grid layer developed on top of this infrastructure to allow for the G-RISICO demonstration is added as annex to the present document in section 7.

The guides and the link to the monitoring web view are available from the project portal www.cyclops-project.eu → left side menu "Links".

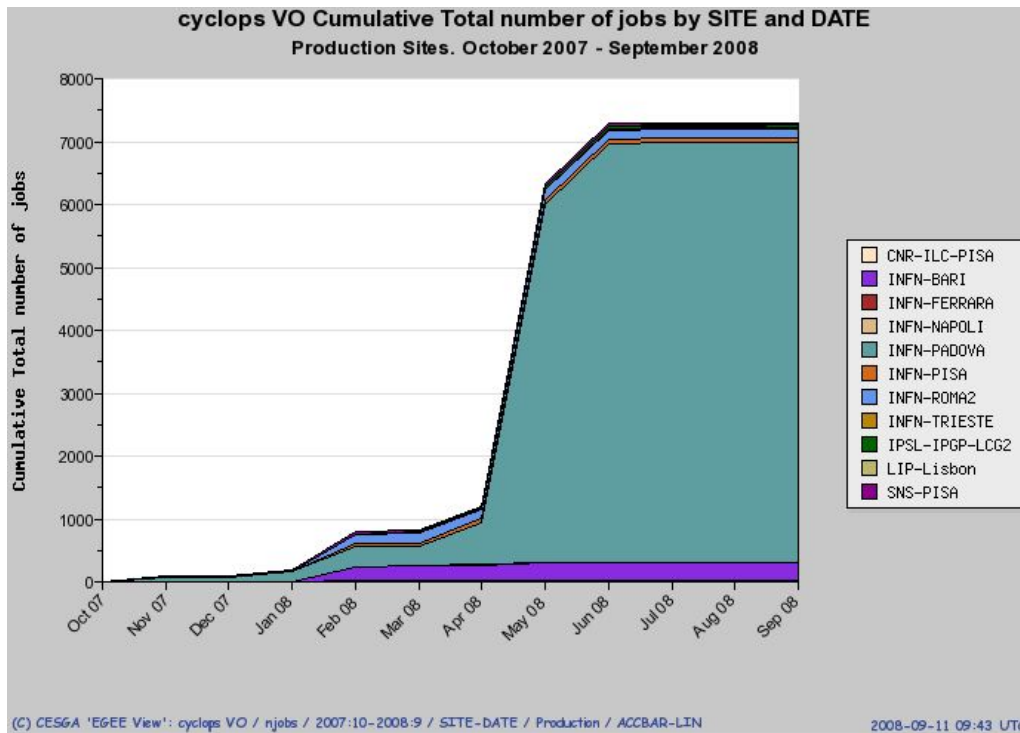


Figure 4-1: cumulative number of jobs vs time for CYCLOPS VO

4.3 CONTACTS WITH OTHER GRID PROJECTS

Following the suggestions of the Technical Expert Group, later outlined also by the EC project reviewers, a number of contacts were established with grid related projects or working groups dealing with issues relevant for Civil Protection and GMES communities, and therefore in common with the CYCLOPS project.

The following initiatives/projects/working groups were involved:

- the OGC-OGF initiative, started in 2007, with the goal of identifying and producing standard, grid-enabled geospatial information tools in the context of a service-oriented architecture, most mature OGC standards being the Web Coverage Service (WCS), the Web Processing Service (WPS), the Web Map Service (WMS) and the Web Feature Service (WFS). CYCLOPS contributed since the beginning presenting its preliminary results on a grid-enabled OGC-WCS at OGF20 of Manchester in May 2007, and at OGF23 of Barcelona in June 2008;
- the GENESI – DR project, started in January 2008, with the goal of creating a standardized approach to implement OGC Web Services (OWS) on top of grid infrastructures, starting from the architectural framework and the prototypal library for implementing OGC-WCS and OGC-WPS on gLite middleware developed by CYCLOPS;
- the DORII project, started in February 2008, with the goal of discussing a reference implementation of the Sensor Observation Service, to be compliant with the OGC standards, and its relationship with the Instrument Element developed within DORII Joint Research Activity. The activity carried out in this context by the University of Minho CYCLOPS partner is described in the annex III in section 8;
- the newly formed “Earth Sciences cluster” within the NA4 activity of the EGEE-III project, started in May 2008, with the goal of providing Civil Protection specific inputs and requirements toward Earth Sciences community and grid services definition;
- the EELA-2 project, the second phase of the EELA EU funded FP6 project aiming to build a bridge between consolidated e-Infrastructures in Europe and the emerging ones in Latin America. The relationship with

CYCLOPS was established by University of Minho partner, who is also involved in EELA-2 project, and its first results are reported in the annex III in section 8.

Two Memorandum of Understanding (MoU), one between CYCLOPS and GENESI-DR and one between CYCLOPS and DORII projects, have been signed by the corresponding project and technical managers. The main aims of the MoUs are:

- to define the expected benefits for the collaboration
- to identify the key individuals of the projects who will take responsibility for achieving the goals described in MoU
- to identify the areas of cooperation, including the specific requirements and resources from the two sides for supporting the cooperation
- to define a schedule and significant milestones for the cooperation

5 CONCLUSIONS

The main goal of the CYCLOPS project is *“to bridge the gap between Grid and GMES communities making Civil Protection people aware of the services provided by Grid infrastructures, and, at the same time, letting Grid researchers to be aware of Civil Protection specific requirements and service enhancements needs”*.

Inter-dissemination between these communities was therefore identified as a fundamental activity, reflected in the project’s organisation by defining a dedicated Work Package: WP2 – Coordination with EGEE activities.

The present document reported the work done in the WP2 context throughout the whole lifetime of the project to achieve this project goal.

The training activity for explaining the EGEE grid infrastructure and its middleware to Civil Protection community exceeded the committed one at the beginning of the project, being the INFN Application’s School added to the initially planned two CYCLOPS training workshops.

The dissemination of CYCLOPS results towards the Grid community was carried out giving presentations and sometime live demonstrations at 13 different grid-related international conferences and workshops.

The CYCLOPS Virtual Organisation has been established and supported from 12 EGEE production grid sites belonging to 3 different countries with an active role in the CYCLOPS project. These grid sites have provided computing and storage resources for developing a prototype of grid-enabled geospatial services later used for the G-RISICO live demonstration.

Strict contacts with other grid-related projects or initiatives having different objectives but sharing common issues with the CYCLOPS project were established. The goal is to provide and receive feedback on Civil Protection matters, joining the efforts in order to define and plan the future European Grid Platform for Civil Protection.

6 ANNEX I: REPORT OF INFN TRAINING COURSE ON PRACTICAL ASPECTS OF APPLICATION'S INTEGRATION IN GRID

6.1 APPLICATION IDENTITY CARD

Application Name: Fire risk assessment and coordination

Acronym: RIS.I.CO.

Application's domain of interest: Environmental Monitoring

Involved Foundations: CIMA Foundation, INFN-Padova, CNR-IMAA

Working group: Mirko D'Andrea – CIMA Foundation (mirko.dandrea@cima.unige.it), Stefano Dal Pra – INFN-Padova (stefano.dalpra@pd.infn.it), Valerio Angelini – CNR-IMAA (angelini@imaa.cnr.it)

6.2 APPLICATION GENERAL DESCRIPTION

RIS.I.CO. is a prevision system designed for the assessment of the distribution of dynamic wild land fire risk over the whole Italian territory which has been developed by the CIMA Foundation for the Italian Civil Protection Agency.

Such an assessment takes place on the basis of static information concerning the vegetation cover and the topography of the territory and of dynamic information consisting in the meteorological forecast, over a certain time horizon (72 hours), provided by a Limited Area Model.

RISICO performs his prediction about potential dangerousness over a discretized domain of 1 km cells (these are about 330.000) covering the whole Italian territory. The model is intrinsically parallelizable being that computation only takes into account cell status without considering correlations between them.

Input data for the model are provided by a network of meteo sensors as well as atmospheric models (Local Area Model). These data gets daily retrieved and aggregated.

RISICO also needs a static set of information describing cells respect to orographic and vegetation characteristics for the discretized domain. The adopted resolution for cells influences accuracy for static variables represented by the territory portion covered by cells. This has a strong impact especially for vegetation, being that Mediterranean regions have a great variety of different coexisting species as also a great importance in determining the fire risk estimation.

RISICO provides its computation output as a set of maps representing predicted behaviour in space and time for those variables which are useful in identifying risky situations. At present these variables indicates potential fire risk, i.e. the danger level of a fire, without considering probability of it to begin.

Output data produced for the whole Italian territory takes about a daily amount of 100MB, with an overall 40MB of needed input data, when using 1km cells. Needed computation time is in the order of 40 minutes using a common workstation.

6.3 PORTING WORK DESCRIPTION

6.3.1 Goal

As we said already, good quality estimation strongly depends on properly sized cell. While the 1km cell size is a good one in terms of needed computation time and storage resources, running the model with 100m sized cells would provide much better estimations, but this would lead to increase a hundred of times both computation time and needed storage space.

Our goal is to run the RISICO model on grid in order to achieve in a reduced time the results for simulation on 100m sized cells and manage produced output in an efficient way. The “reduced response time” is a quite important requirement for Civil Protection operators in order for them to be effective.

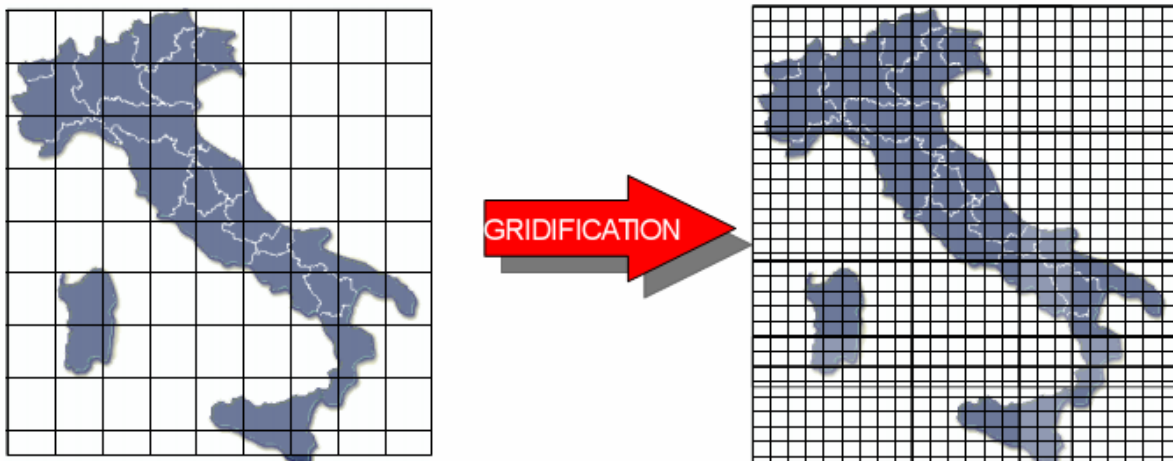


Figure 6-1: accuracy is improved reducing territory cell size from 1km to 100m

6.3.2 Grid porting strategy

Thanks to the intrinsic parallel nature of RISICO, our first considered approach has been that of adopting a “divide et impera” strategy, which basically consists on submitting N RISICO instances, each one of them working on subsets of cells obtained by partitioning the whole set. A proper range choice for the N value permits more satisfying results. As a rule of thumb, too high values may increase queue times effects or the probability that a (single) job encounters a problem (i.e. it gets queued on a busy site with high wait time), while too little values obviously tend to reduce benefits of distributed computation. Experiments with N in the range 10, 30 have shown satisfying results.

6.3.3 Description

Global RISICO output is obtained as result of many jobs, each one computing a given portion of the whole domain. Jobs are identical and independent.

Every job makes use of a set of files which are available into one or more Storage Element (SE) and identified through their Logical File Name. These files are copied into the closest SE by a python script running on the User Interface.

They are:

- `celle_nn.tar.bz2`: defines computation domain for the job numbered nn with static information (cells description).



- `input_data.tar.bz2`: contains information related to input variables (meteo data, dynamic). This file is common to all jobs;
- `output_nn_data.tar.bz2`: this is the output file produced by the job numbered `nn`;
- `stato0_nn.tar.bz2`: status information for the job numbered `nn`. After computation, a new status file is produced, which will replace the old one and will be used in the next RISICO run;

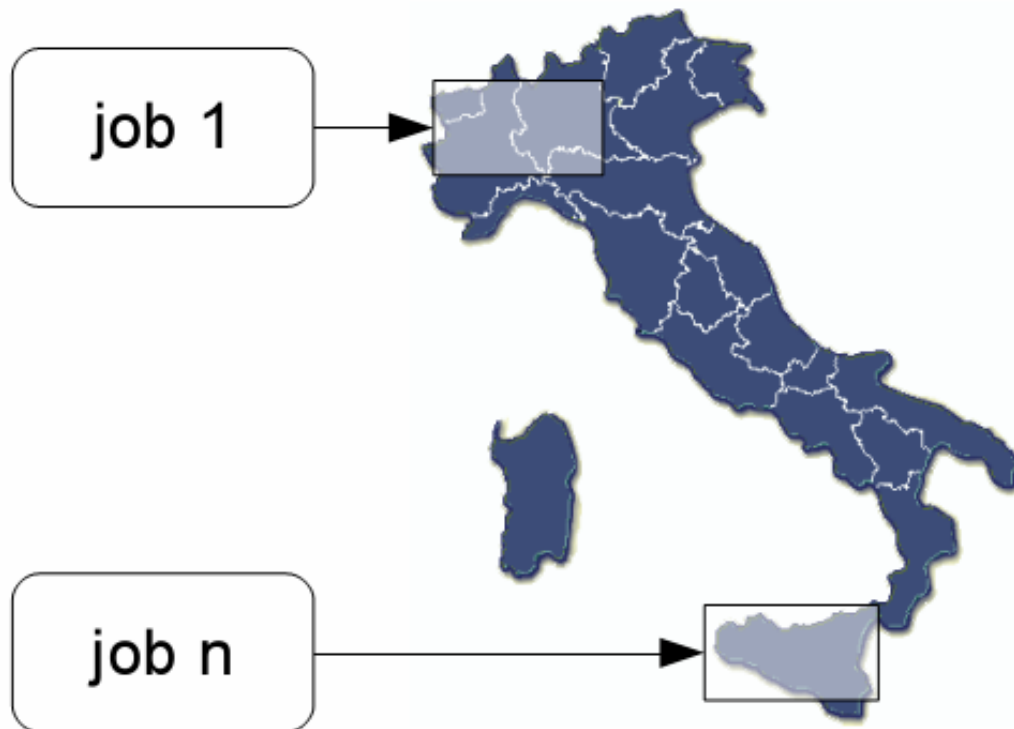


Figure 6-2: each job is processing different portions of territory

A fundamental piece for the Gridified RISICO system is the `gJobSubmitter` module, which takes care of setting up all needed things for jobs to run, basing on preferences specified in a configuration file. This module, when launched, uploads files into the Storage Elements, creates a JDL file for each job and submits the jobs itself. Then it follows their execution until it terminates, retrieving produced output for each successfully terminated job.

Two methods are used by the `gJobSubmitter` module to keep track of the jobs status: a “direct control”, by regularly polling the Logging & Bookkeeping server (`glite-wms-job-status`) and an alternative one, which consists in evaluating the presence on the SE for the expected output file (which LFN is known in advance). Quite often this permits to detect job's completion and retrieve its output file before it is “officially” sentenced to be finished by the LB server, and this saves a latency time who can quite commonly be in the order of a five to ten minutes.

Also with the intent of reducing latency times, it is optionally possible to specify in the configuration file one or more queues where jobs are to be directly submitted. This may permit to skip the MatchMaking phase in the Workload Management System. This can be useful to have jobs submitted on queues which are a-priori known to be quickly available and offer a low latency service. It may happen that despite having specified a “lowest response time” as a requirement in the JDL file, the job is still delivered on a busy site or in a temporarily inefficient one.

In order to reduce the impact of this kind of problems it is possible to resubmit a job which is still waiting after another one has finished already to the same queue used by the successful one, which has proven to be effective.

6.3.4 gJobSubmitter Module

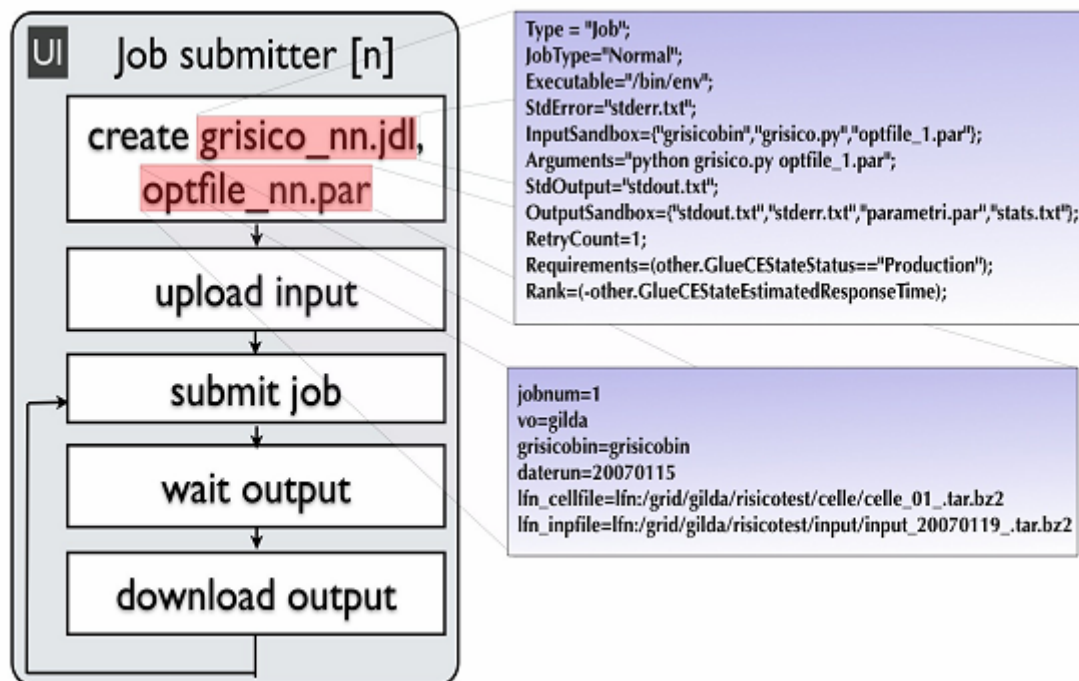


Figure 6-3: the job submitter component schema

Operations accomplished by the gJobSubmitter module are:

- Copy input files on a close SE using a lcg-cr call;

Example: `lcg-cr --vo cyclops file:/home/user8/grisco/input.tar.bz2 -l lfn:/grid/risco/input/input_20071127_tar.bz2 -d prod-se-02.pd.infn.it`

- Creation and submission of n independent jobs;

Example: `glite-wms-job-submit -a -o jid_20071127.txt grisco_1.jdl`

- Instantiation of n control processes JobSubmitter and polls job termination;
- Output file retrieval for completed jobs.

6.3.5 Final considerations on the adopted submission strategy

In Civil Protection context, a “reduced response latency” is to be preferred respect to a “high productivity in the medium average period”. This is why the chosen submission strategy may optionally skip non strictly needed phases, such as MatchMaking in the WMS and “official” job status check from the LB point of view. A further potential advantage is that the number of “point of failure” gets reduced: if a component is in a critical condition, the ability to be robust respect to his undesirable behaviour may be important. Practical performed tests have effectively proven the ability to retrieve job output minutes before LB declared it has done. Also, the job submitter module can decide to resubmit a still waiting job on a provenly effective queue.



REF:CYCLOPS-WP02-D17-INFN-002

Issue: 01 Rev: 00

Date: 24/11/2008

6.4 PERSONAL COMMENTS

This “application porting school” has given us a great opportunity to learn about grid architecture and to practice with its available instruments so to test what useful potential it offers for civil protection issues.



7 ANNEX II: DOCUMENT DESCRIBING THE GRID LAYER PROTOTYPE DEVELOPED TO REALIZE THE CYCLOPS DEMONSTRATION

7.1 TERMINOLOGY

CYCLOPS	Cyber-infrastructure for CiviL protection Operative Procedures
RISICO	RISchio Incendi e COordinamento
TA	Technical Annex
WP	Work Package
GILDA	Grid INFN Laboratory for Dissemination Activities
GMES	Global Monitoring for Environment and Security
EGEE	Enabling Grid in E-science
CA	Certification Authority
CE	Computing Element
RA	Registration Authority
RB	Resource Broker
LRMS	Local Resource Management System (batch system)
VO	Virtual Organization
VOMS	Virtual Organization Membership Service
SVN	Subversion.
WM	Workload Manager
WMS	Workload Manager Service
WN	Worker Node
WCS	Web Coverage Service
WPS	Web Processing Service
WAR	Web ARchive
Publisher	Web service sub-module accepting HTTP requests about instantiated resources.
Wrapper	A software component who wraps execution of a given program .
JSON	JavaScript Object Notation
GUI	Graphical User Interface

7.2 WORKFLOW OUTLINE

This Section briefly describes the interaction between software components and services involved in a demo run of the use case selected application RISICO.

7.2.1 Involved components

7.2.1.1 Graphical User Interface (selection and submission)

Once launched the CYCLOPS GUI (developed in Java) on a common client workstation, a map of the Italian region is presented to the user, who can select a portion of interest for a run of the RISICO application, together with desired

settings (such as cells size and prediction time range, desired Grid jobs number or execution time) and submit the task to the other involved components for execution. During the task execution phase, the current status of the overall process is available from the GUI itself.

7.2.1.2 WPS, WCS (Extraction from coverage and input data setting)

The Web Processing Service receives XML requests from the GUI. The WPS takes charge to ask the Web Coverage Service in order to extract needed input data for the RISICO application from one or more coverages. These coverages are stored in Grid Storage Elements or locally (other non Grid data sources may be defined as well). In the case of the described demo the coverage are stored on SE and extracted data are also stored on SE. This step is thus performed by mean of Grid jobs submitted by the WCS. While running, each Grid job posts web messages on a Publisher component thus reporting information about advancement status. This information is made available in real time on the GUI and is used to inform the user about current status of the overall execution.

In current software deployment, the WPS and WCS services are running with tomcat5 and java 1.5 on standard SL4.x servers. All installed software comes from standard rpm repositories except of course for WPS and WCS which are provided as one java WAR file each. Needed disk space after installation is in the order of 20 Mb each. Both services may well be running on a single machine with common hardware performances. For the comfort of developing and testing however, they have been deployed on two separate machines. The Publisher component is installed on the same machine that runs the WCS.

7.2.1.3 WPS, Wrapper (submission and wrapping of RISICO jobs)

Once the previous step is completed and the input files needed for RISICO execution are properly stored in the SEs, the WPS creates needed JDL job description and an additional configuration file, encoded in JSON format.

Then the WPS submits the jobs to the Grid for the execution of the RISICO simulation.

In order to facilitate the input and output phase a python script (Wrapper) was developed that handles execution of the real executable in the WN according to parameters specified in the additional configuration file provided through the InputSandbox. When a job starts, the Wrapper keeps posting http messages from the WN to the Publisher, describing each status advance or reporting any error condition that may arise at runtime.

7.2.1.4 Graphical User Interface (workflow monitoring and output visualization)

The GUI presents final output in graphical form to the user once everything is done and from then on the user has the ability to select a desired specific representation (i.e. humidity map, fire risk map) from the available ones.

7.3 THE WRAPPER

Although grid jobs are specified by JDL syntax and actually executed by a so called “Job Wrapper” which is an intrinsic part of the Grid operational model, a few operations still are to be explicitly implemented by the user, such as downloading files to be computed in the WN. Moreover, some steps, such as creating a correct java command line are more easy to be created when running in the actual environment (the WN) rather than while creating the JDL file for the job, when certain conditions can only be predicted or “guessed”. Because of this and other reasons (see “motivation” section below) we choose to define and write our own wrapper, with a quite flexible behaviour specified by in a specific configuration file. Please note that specifications and requirements in the JDL are not overlapping with those expressed in the configuration file. Information useful for the matchmaking process by the WMS are provided in the JDL, while other operational steps to be performed during actual execution are specified in the JSON file and accomplished by our wrapper. This way we separate grid specific requirements and execution specific ones in two separate files.

The wrapper is a python program which offers a few classes providing utilities used to perform various service operations, such as create on the Worker Node a proper execution environment just before launching the actual application, inform at runtime an external web service (the Publisher) about relevant events as long as they occur and finally to prepare output files in a convenient format (such as a compressed archive) and upload them on SE or to the Publisher.

The behaviour of the wrapper is quite flexible and is specified in a JSON configuration file, which is created by the WPS and provided through InputSandbox.

7.3.1 Builtin capabilities

The wrapper has built-in capabilities to perform these basic tasks:

- download files from SE or from web services (act as HTTP client, method POST).
- compress / uncompress files
- archive / unarchive files and directories
- upload files to SE or web services (act as HTTP client, method PUT)
- proper start of application
- timing, events and error log
- signalling events and errors to the Publisher (act as HTTP client, method PUT)

Other capabilities are easy to implement and add to the wrapper if needed. Once done, they may be used by specifying them in the JSON file. Next section will provide further details.

7.3.2 Configuring wrapper with JSON

The actual behaviour of a wrapper instance is specified in a `conf.json` file which defines data objects, parameters and actions to be taken in time order. JSON syntax is equivalent to the python syntax for dictionaries and the python code in the wrapper access this dictionary to retrieve sequences of actions to perform.

Worth to be mentioned objects and parameters specified through JSON are:

- **InputFiles:** a list of specifications indicating which files are to be downloaded (web url or Logical File Name, SURL, TURL for files in SE) and eventually which actions are to be taken after downloading the file (unzip, untar) and before starting the application and the local name for the downloaded file
- **OutputFiles:** similar to the previous specification, but actions are performed in reverse order (archive, compress, upload ...)
- **Executable:** a set of specifications telling which executable has to be executed, where to find it and where to check for needed libraries. This information is used to prepare the correct command line (it permits for example to specify a java version newer than the default available one and to customize libraries paths).
- **Url:** used to specify the publisher baseurl where runtime messages are sent.
- **JobIdentity:** this defines *runId* and *jobnum* values. These are needed to the other software components to distinguish the jobs. The *runId* completes the url used to contact the Publisher and must be known by the publisher itself to be valid.
- **RisicoVariables:** The RISICO application itself reads working parameters expressed with JSON syntax. In general, a single `conf.json` file may specify configurations both for the wrapped application and the wrapper itself.

Depending on the wrapped application, any of the described objects may or may not be mandatory.

7.3.3 Motivation

Reducing time latencies and being robust respect to faulty situation may be a major issue in many situations. Having runtime information from jobs running in the WN permits to reduce unnecessary wait time and to eventually skip requests to Grid components (such as the Logging and Bookkeeping service) thus avoiding a possible point of failure.

Moreover, improvements in the interaction model between Publisher, WPS and WCS may permit to further reduce time delays, by adding capabilities to the Publisher: for the sake of example, when a “Job finished, output uploaded” message is received by the Publisher from a given WN, a new `conf.json` may be given as answer to the wrapper (still running in the WN) which could then start computing a new piece of data with no submission delay at all. This would also permit to effectively run “urgent” tasks first (as soon as a job starts on a WN) even though that usually no warranty exists for two jobs submitted at the same time about which one is going to start execution first.

7.4 LAYER TO ACCESS TO GRID INFRASTRUCTURE

Grid computing offers a model for solving massive computational problems using distributed resources (CPU cycles, disk usage, etc.) of large numbers of different organizations. So for the CYCLOPS demo using the Grid is a requirement.

To simplify the access to the Grid functionalities (submission and control of jobs) needed for the demo an ad hoc layer has been developed.

This Grid layer is compound of the following main classes:

- eu.cyclops_project.grid.JobManagementTool
- eu.cyclops_project.grid.JobSubmissionHelper
- eu.cyclops_project.grid.ConfigurationFile

From now on, the class names will be used with the short name instead of the full name.

In the next paragraphs we'll see in detail the Grid layer API implemented.

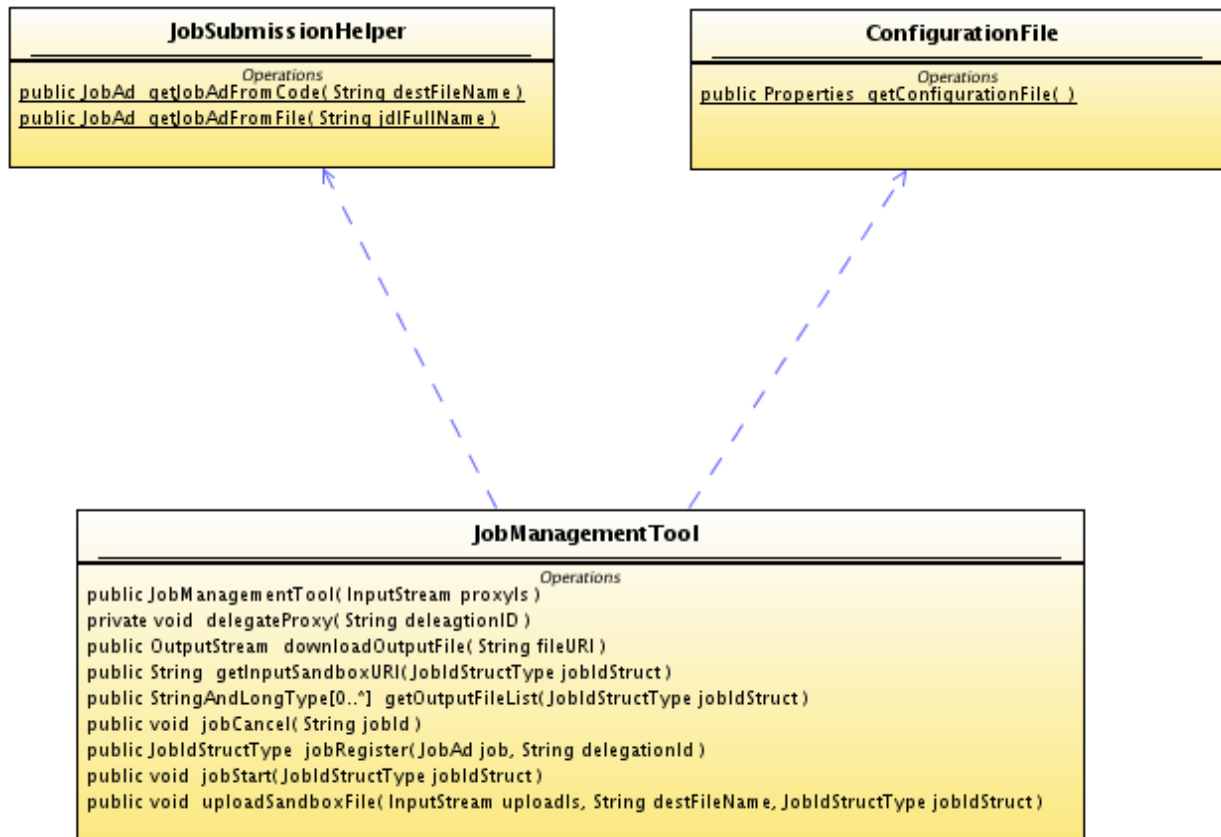


Figure 7-1: Grid Layer's class diagram

7.5 JOBMANAGEMENTTOOL CLASS

The JobManagementTool class uses the WMProxy client API for accessing the Grid resources. WMProxy is a component of gLite Workload Management System (WMS) that is responsible for accepting incoming requests and provides support for the job control functionality through a Web Services based interface.

7.5.1 JobManagementTool API

JobManagementTool { From grid }
<i>Operations</i>
<pre> public JobManagementTool(InputStream proxyIs) private void delegateProxy(String delegationID) public OutputStream downloadOutputFile(String fileURI) public String getInputSandboxURI(JobIdStructType jobIdStruct) public StringAndLongType[0..*] getOutputFileList(JobIdStructType jobIdStruct) public void jobCancel(String jobId) public JobIdStructType jobRegister(JobAd job, String delegationId) public void jobStart(JobIdStructType jobIdStruct) public void uploadSandboxFile(InputStream uploadIs, String destFileName, JobIdStructType jobIdStruct) </pre>

Figure 7-2: JobManagementTool class

7.5.1.1 JobManagementTool(InputStream proxyIs)

First of all, you should acquire a proxy certificate. This can be achieved through the command line interface of a UI machine (with *voms-proxy-init* command). After that, you can call the constructor of the JobManagementTool class.

The argument proxyIs is an InputStream object of the proxy certificate file.

The constructor after having read the CYCLOPS' configuration file and used the information present in it, references a WMProxyAPI object and creates a GSSCredential object that it will be used to authentication to GridFTPClient for uploading and downloading files to/from WMS.

GSSCredential is an interface that encapsulates the GSS-API credentials for an entity. A credential contains all the necessary cryptographic information to enable the creation of a context on behalf of the entity that it represents.

7.5.1.2 JobIdStructType jobRegister(JobAd job, String delegationId)

This method registers a job for submission. The unique identifier assigned to the job is returned to the client. This operation only registers the job and assigns an identifier to it. The processing of the job (matchmaking, scheduling etc.) within the WM is not started. The job is "held" by the system in the "Registered" status until the jobStart operation is called.

Since the WMS providing some services needs to interact with other services, operating on behalf of the user, a delegation process is needed to transfer client proxy credentials to the server host.

The jobRegister method also performs the delegation process using the specified user delegationId (see delegateProxy method).



7.5.1.3 **uploadSandboxFile(InputStream uploads, String destFileName, JobIdStructType jobIdStruct)**

File upload is performed by GridFTP server available on the WMS node. The connection to the GridFTP server is established performing the authentication with specified user credentials (GSSCredential object created previously).

The destination URI associated to the job, identified by the jobId, where the job input sandbox files can be uploaded by the client is retrieved, using the *wmProxyApi.getInputSandboxURI* method. The location is created in the storage managed by the WM.

7.5.1.4 **jobStart(String jobId)**

This method triggers the submission of a previously registered job. It starts the actual processing of the registered job within the Workload Manager. It is assumed that when this operation is called, all the work preparatory to the job (e.g. input sandbox upload, registration of input data to the Data Management service) has been completed previously.

7.5.1.5 **jobCancel(String jobId)**

This method cancels a previously submitted job identified by its jobId. If the job is still managed by the WM then it is removed from the WM tasks queue. If the job has been already sent to the CE, the WM simply forwards the request to the CE.

7.5.1.6 **StringAndLongType[] getOutputFileList(JobIdStructType jobIdStruct)**

This method returns the list of URIs where the output files created during job execution have been stored in the WM. Of course file retrieval is only allowed when the job has finished its execution (either successful or failed).

7.5.1.7 **OutputStream downloadOutputFile(String fileURI)**

File download is performed by GridFTP server available on the WMS node. The connection to the GridFTP server is established performing the authentication with specified user credentials (GSSCredential object created previously). The URI of the file to be downloaded is retrieved using the *getOutputFileList* method. Of course file retrieval is only allowed when the job has finished its execution (either successful or failing).

7.5.1.8 **void delegateProxy(String delegationId)**

The delegation is the process used to transfer rights and privileges to another party. Before invoking services such as submission, a delegation process is needed to transfer client proxy credentials to the server host. Delegated credentials are uniquely identified by the association of the delegation identifier, provided by user, and the user's DN within the credentials.

Delegating credentials is done in the *delegateProxy* method with the two following calls:

```
proxyReq = wmProxyAPI.grstGetProxyReq(delegationId);
```

Gets a proxy identified by the delegationId string.

```
wmProxyAPI.grstPutProxy(delegationId, proxyReq);
```

The created proxy is sent to the server and identified by a delegationId string. This string can be used to call some services accepting a delegationId string as input parameter (like jobRegister, jobSubmit, etc) until its expiration time.

7.6 JOBSUBMISSIONHELPER CLASS

JobSubmissionHelper is an utility static class useful to make simple the creation of a jobAd object from a JDL description. A JobAd object provides a representation of the job description in the JDL language (fully compliant to WMS specification) and the functions for building and manipulating it. Afterwards a JobAd object is been created it can be submitted in the WMS.

The JobSubmissionHelper class publishes two methods:

- *JobAd getJobAdFromCode(String destFileName)*
- *JobAd getJobAdFromFile(String jdlFullName)*

7.6.1 JobSubmissionHelper API

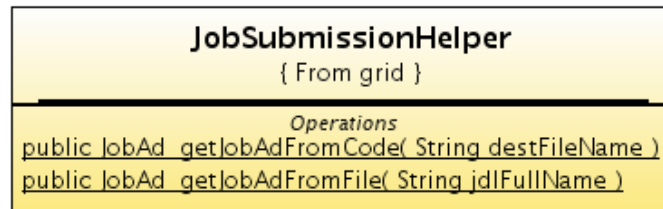


Figure 7-3: JobSubmissionHelper class

7.6.1.1 JobAd getJobAdFromCode(String destFileName)

This method can be only used if you want to build the JobAd object from code. If that is the case, you should override the method. The method has been implemented for a specific JDL description expecting a unique input file.

The argument *destFileName* is the name of the unique input file specified in the JDL description.

7.6.1.2 JobAd getJobAdFromFile(String jdlFullName)

This method is of more general use. In fact, it allows users to build a JobAd object from a generic JDL description.

The argument *jdlFullName* is the full absolute path of the JDL file.

7.7 CONFIGURATIONFILE CLASS

ConfigurationFile is an utility static class to access to the CYCLOPS' configuration file. It has been developed according to the Singleton pattern and publishes a unique method returning a Properties object for accessing to the configuration file:

- *Properties* `getConfigurationFile()`

7.7.1 ConfigurationFile API

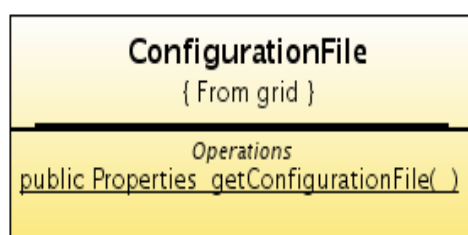


Figure 7-4: ConfigurationFile class

7.7.1.1 Properties getConfigurationFile()

This method returns a Properties object in order to simplify the access to the information included in the CYCLOPS' configuration file (see next paragraph).

7.7.2 The CYCLOPS' configuration file

The following is the CYCLOPS' configuration file.

```

# URL of the wms proxy
wms_wmproxy_server=https://prod-wms-01.pd.infn.it:7443/glite_wms_wmproxy_server
#wms_wmproxy_server=https://glite-rb-00.cnaf.infn.it:7443/glite_wms_wmproxy_server
# points to the ca certificate(s).
certDir=C:/Documents and Settings/admin/.globus/certificates
  
```

where:

`wms_wmproxy_server` is the URL at which the WMPProxy service is available

`certDir` is the location (directory) where to look for the CA (Certificate Authority) certificates.



7.8 HOW TO USE THE GRID LAYER

In this chapter we can see an example of use of the Grid layer described so far. For more examples and details, you can see the Junit test class `JobManagementToolTest` included in the software release.

In order to execute a job, the following steps must be performed:

- instancing a `jobManagementTool` object
- creating the job from JDL
- submitting the job
- retrieving the output

7.8.1 Instancing a `JobManagementTool` object

First, an instance of the class `JobManagementTool` must be made. The constructor for this class is:

```
JobManagementTool jobManagementTool = new JobManagementTool(proxyIs);
```

The argument *proxyIs* is a `FileInputStream` object of the proxy certificate file. The user must create his own proxy certificate before instancing the `JobManagementTool` object.

7.8.2 Creating the job from JDL

The creation of the job is done with the following call on the `JobSubmissionHelper`:

```
job = JobSubmissionHelper.getJobAdFromFile(jdlFullPathName);
```

The argument *jdlFullPathName* is the full absolute path of the JDL file describing the job.

7.8.3 Submitting the job

The submission of the job is split into these phases:

- Register the job with the user's `delegationId`
- Upload the input files
- Start the job

First, the job is registered with the user's `delegationId` by a call to the `jobRegister` method of `JobManagementTool` object.



```
JobIdStructType jobIdStruct = jobManagementTool.jobRegister(job, delegationId);
```

The argument `job` is a `jobAd` object and the argument `delegationId` is the user's delegation identifier submitting the job. The result of the method call is an object of type `JobIdStructType`.

The completion of the the `jobRegister` phase means that the job has been taken into account within the Grid environment and is ready to be actually submitted to Grid resources. All auxiliary actions for the job to run successfully on the remote resource can be performed in the phase between job register and job start.

So now, the input local files needed by the job at run time have to be transferred to the WMS. To do that, the following should be done:

```
InputStream uploadIs = new FileInputStream(inputFileFullPathName);  
jobManagementTool.uploadSandboxFile(uploadIs, destFileName, jobStruct);
```

This segment of code should be repeated for every input local file specified in the job input sandbox section of the JDL file. At this point, the job can be started easily with the call to `jobStart` method:

```
jobManagementTool.jobStart(jobStruct);
```

7.8.4 Retrieving the output

Once the job has finished, the output produced can be retrieved. First, we should acquire the URLs of the files on the WMS machine and then using the `downloadOutputFile` method to download them to the local machine.

```
StringAndLongType[] list = jobManagementTool.getOutputFileList(jobStruct);
```

The result of the method call is the name (URL) list of the output files to be downloaded. Once the file's URL is known, we can copy it from the WMS to the local machine:

```
OutputStream os = jst.downloadOutputFile(fileUrl);
```

This call should be repeated for every output file.

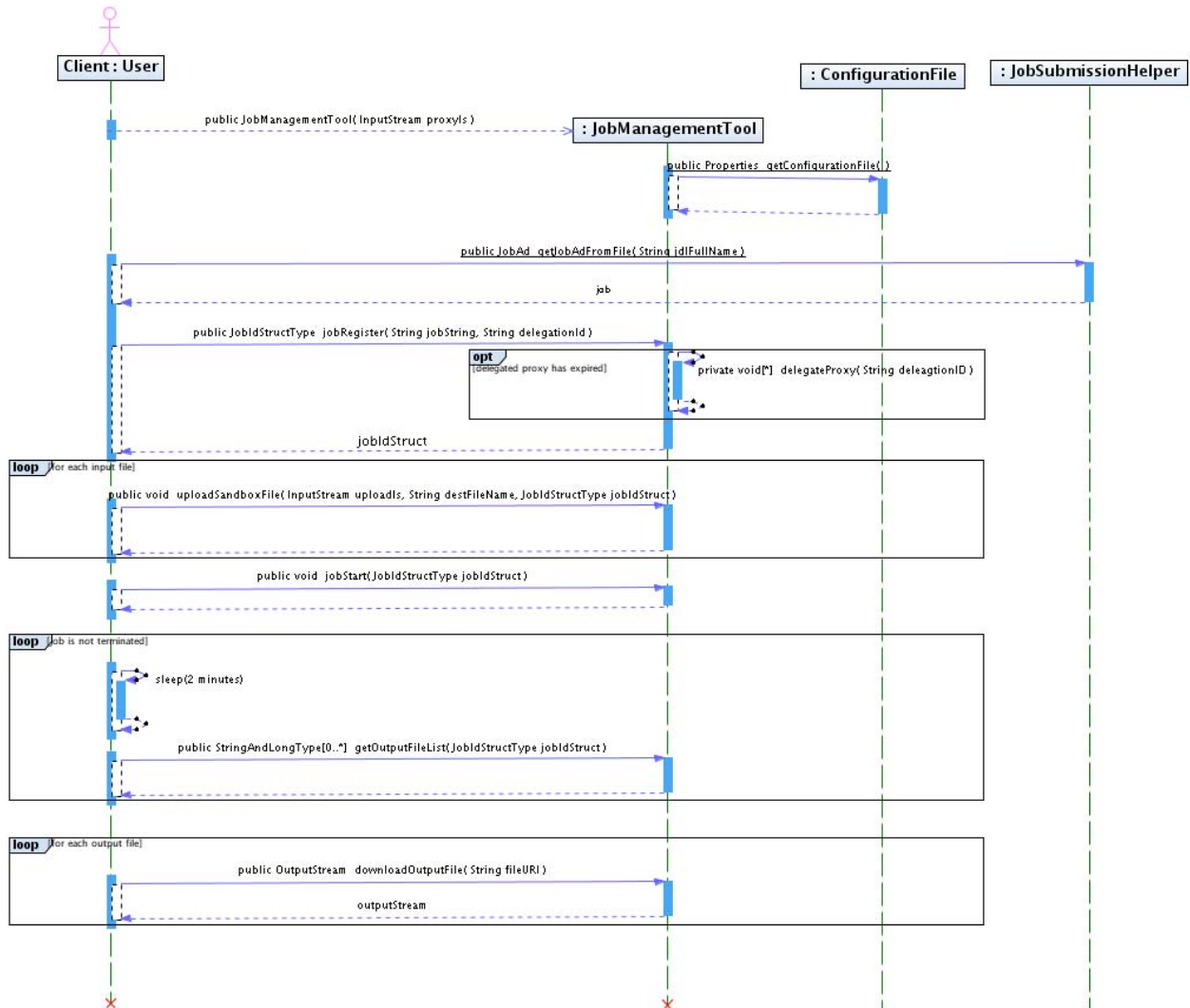


Figure 7-5: sequence diagram: how to use the grid layer

7.9 SOFTWARE DOWNLOAD

The software is stored in a Subversion repository (SVN) at the following URL:

<http://ulisse.pin.unifi.it/svn/cyclops/branches/collaborative>

Important note: To access to the software you need to ask for an SVN account to the SVN CYCLOPS administrator.

The build system of the software is based on the tool Maven2. So if you want to build the grid layer or execute the test class (JobManagementToolTest) you need before downloading and installing the tool Maven2 [3].



REF:CYCLOPS-WP02-D17-INFN-002

Issue: 01 Rev: 00

Date: 24/11/2008

7.10 BIBLIOGRAPHY

- [1] EGEE Global Security Architecture for web and legacy services EU Deliverable DJRA3.3, EGEE-JRA3-TEC-602183-DJRA3.3-1.0, <https://edms.cern.ch/document/602183>
- [2] WMPProxy guide: <https://edms.cern.ch/document/674643/1>
- [3] Maven2: <http://maven.apache.org/>

8 ANNEX III: UNIVERSITY OF MINHO CONTRIBUTION

The amendment of December 2007 incorporated two new partners in the original consortium. After the period of integration, University of Minho clearly identified a number of initiatives to establish stronger collaborations with others partners ARMINES-EMA, other related EU projects - Int.EU.Grid, EELA-2 and DORII - and grid initiatives - Portuguese National Grid Initiative - and engage actively with them to promote and enhance CYCLOPS results.

Following the recommendation that results and outcomes from meetings and activities should be explicitly mentioned and detailed for evaluation, dissemination and re-use, next we briefly described review major University of Minho contributions.

8.1 GRID INFRASTRUCTURE

CYCLOPS first review report stated that “WP2 has to keep running and enlarge the CYCLOPS VO infrastructure, deploy the two test cases/prototype applications (Forest Fires and Flash Flood), achieve a significantly higher number of participants from the CP, GMES, Geospatial communities at the training events and provide in a more systematic way feedback to EGEE”.

The above statement assumes that “The EGEE middleware is installed and configured at the Civil Protections sites for test purposes” and that every CP is equipped with “a minimal set of Grid elements needs to be deployed to form a dedicated Grid tested”. However, at that moment, in Portugal there was no infrastructure at the CP organization that can be dedicated to support the test-bed and training events.

To be able to respond to this situation, CYCLOPS amendment annex included a subcontracting item where University of Minho considered to subcontract a service provider for the computation, communication and storage infra-structure needed to install and maintain a minimal EGEE site and support the foreseen dissemination actions, such as: 1) running the selected use-cases application as demos in several GRID/GMES events including the CYCLOPS Mid-term and Final conferences, 2) other planned training sessions.

As a consequence of this possibility, University of Minho deployed a small pilot dedicated Grid test-bed where to install and test the user-case applications in the Grid environment. The site Uminho-CP is based on the up to date existing gLite 3.1 EGEE middleware that consist of a minimal set of Grid elements: UI (User Interface), 1 MON (Monitoring box) , 1 CE (Computing Element), 1 SE (Storage Element), 16 WNs (Worker Nodes). The total computer power achieves the 300GFlops, what is equivalent to 24 WN (Working Nodes).

The site Uminho-CP belongs to EGEE SWE Federation and it also adhered to CYCLOPS VO organization. The infrastructure has been used during the evaluation of the RISICO application porting to EGEE and also during the demonstrations planned for the mid-term and final conferences.

8.1.1 EGEE ROLL

Grid service deployment, in particular for EGEE sites, still lacks complete solutions for the distribution, installation and configuration of middleware to build managed Grid infrastructures.

In this context, University of Minho activity focused on an easy to use framework that manages the installation of all EGEE components with minimal resources, which resulted from the gLite middleware installation and configuration at University of Minho. The objective is to automate the deployment of the EGEE Grid middleware to offer a common operating environment to the geographically dispersed European Civil Protection community.

We developed the EGEE Roll, to provide a standardized software stack across all the sites of a CP Virtual organization, tackling two of the most time-consuming portions of a site installation: the architecture and the mechanisms. The framework enables site administrators to incrementally and programmatically create and modify the graph description for all the appliances (node) types required by the EGEE model such as CE/SE/MON/UI/WNs.

The approach to configure EGEE proved to be pertinent in the context of the development of infrastructures and services for virtual organizations, including software update and user management. It guarantees the interoperability

across sites and full customization at any CP site administrative domain to explore the capabilities of the Grid in CP applications and to reduce the overall time required for the deployment of an EGEE site.

8.1.2 CP Virtual Organization

Figure 8-1 shows a Civil Protection Virtual Organization that consist of a set of physically distributed EGEE sites located within the administrative domain of each European CP participating site, belonging to the same VO, sharing resources and interacting over the Internet.

The architecture model of each site, as can be seen in Figure 8-2, comprises a set of Worker Nodes (left) in a tightly coupled LAN connected by a private Ethernet switch and several different server components (right) connected both to the internal network and to the Internet.

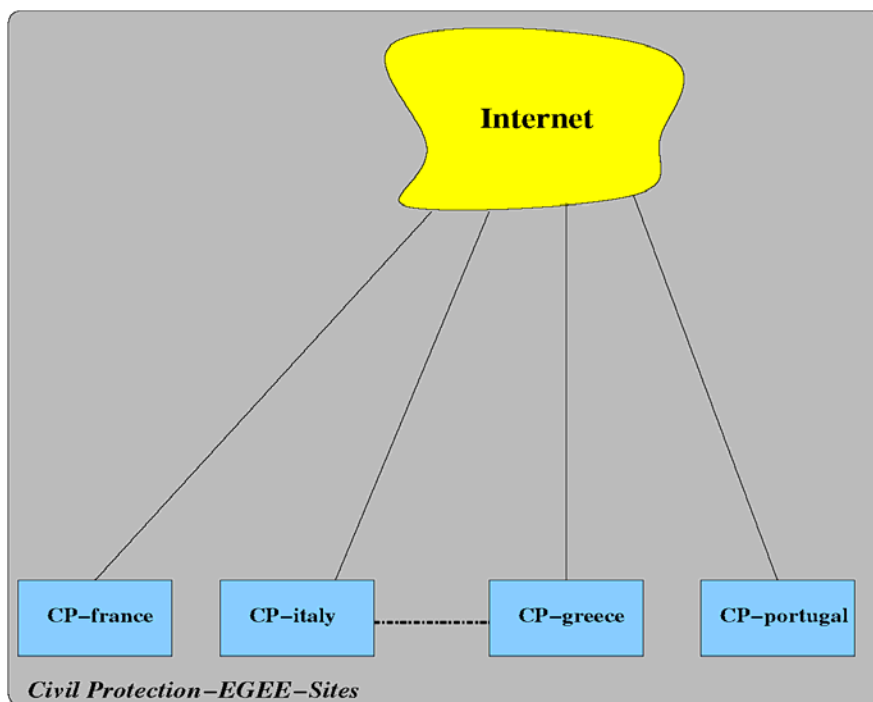


Figure 8-1: CP EGEE Sites

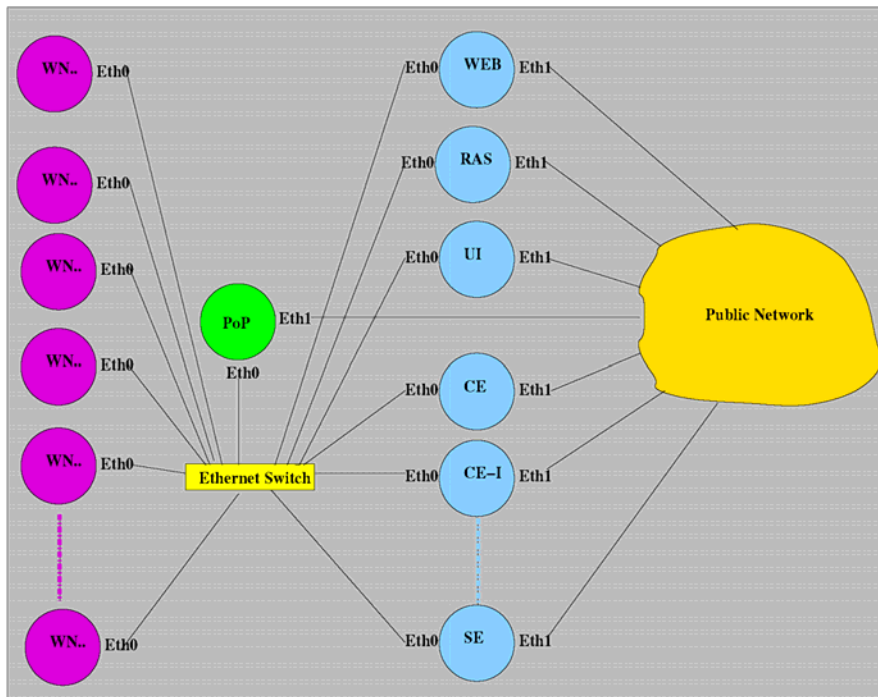


Figure 8-2:EGEE Roll Architectural model

Each site has a variable number of WNs and the minimum set of servers required by EGEE, including: 1) one site BDII; 2) at least one CE and a number of Worker Nodes totalling at least eight CPUs/cores attached to it; 3) at least one SE having a capacity of one 1 TByte or more.

The Front End is a point of presence (pop) of a site that serves as a software repository for all the different types of nodes. It may also be used as a login and compile host to allow local submission of jobs.

The Wide-area kick-start functionality of the Rocks toolkit together with the EGEE roll provide CP organizations a distribution platform and a standardized base software stack across all the sites of the Civil Protection VO.

The common software base allows that enhancements developed in any site can be easily deployed in other partners' site and custom distribution of a site can be transmitted to other site's front ends over the Internet. The approach guarantees the interoperability across sites and full customization in each site's administrative domain.

8.1.3 Conclusions

Using standard tools such as YAIM for installing EGEE sites still involves a significant technical effort. In order to configure the site using the YAIM toolkit, an administrator must choose how to install and configure the grid by manually editing several configuration files and executing the YAIM script.

In the development of the EGEE roll we confirmed that the effort in installing and administrating an EGEE site could be drastically reduced by applying the techniques developed by the Rocks group. Extending the Rocks' approach to configure EGEE replaces the manual software installation and configuration by a fully-automated node installation process, including software update and user management, reducing the administrator's effort and the overall time required to deploy an EGEE site.

Here below is the list of grid related events where oral and/or poster presentations about EGEE ROLL were accepted:

- Pina, B. Oliveira, A. Serrano, V. Oliveira, "EGEE Site Deployment & Management Using the Rocks toolkit", 2nd Iberian Grid Infrastructure Conference, Porto, May 2008

Updated paper: <https://pop.cp.di.uminho.pt/cyclops/wp-content/uploads/2008/09/egee08.pdf>

Presentation: <https://pop.cp.di.uminho.pt/cyclops/wp-content/uploads/2008/09/ibergrid-presentation-amp.pdf>

- Pina, B. Oliveira, A. Serrano, V. Oliveira, “EGEE Roll - A framework to Fully-automated site Deployment & Management”, EGEE’08 Conference, Istanbul - Turkey

Abstract: <http://indico.cern.ch/abstractDisplay.py?abstractId=23&confId=32220>

Poster: <https://pop.cp.di.uminho.pt/cyclops/wp-content/uploads/2008/09/egge-istambul-1.pdf>

8.2 INTEGRATION OF SENSORS

CYCLOPS WP4 Technical Meeting that took place on the 20th-21st of December 2007, (see http://www.cyclops-project.eu/Default.aspx?id_menu=9) at the premises of the University of Florence in Prato focused on the following points:

- WP4 workplan:
 - Geospatial service layer implementation on top of gLite;
 - Porting of Civil Protection applications.
- New Technical Annex and amendment:
 - Definition of the tasks of the new technical partners (EMA and UMINHO).

The initial part of the discussion involved all the participants analyzing in depth issues, such as:

- The need to clarify the advantages in using the Grid environment for CP applications for the two considered use-cases;
- General approaches to the parallelization of computing tasks for CP applications.

The final part involved small groups focused on specific technical aspects:

- How to port SPCGD on gLite using the same approach planned for RISICO, considering the decision to use Grid-enabled OGC services in the SPCGD framework;
- The problem of data formats considering the possibility of porting input and output data to a common format (geoTIFF, netCDF?);
- Possible approaches on the implementation of sensor access services.

After the discussion several tasks were proposed, in particular University of Minho was committed to:

- enable a working node in the CYCLOPS VO;
- work on the access to grid-enabled sensors.

8.2.1 Approach

The goal of CYCLOPS project was to join Grid and GMES communities in order to fully exploit the Grid capabilities for GMES applications. One of the scenarios GMES had was the existence of remote wireless sensor networks (RWSN), which could monitor and transmit real-time data from those remote places, like temperature, humidity or wind speed, in order to prevent or react more accurately to situations of natural disasters like floods, fires or earthquakes.

In this context University of Minho evaluated the possibility of using the middleware called GRIDCC, now evolving as DORII, to control, monitor and collect data from a RWSN. The core and novel element of the GRIDCC/DORII middleware is the Instrument Element (IE), which offers a Web service interface to integrate instruments and sensors into the Grid. The second key component is the Virtual Control Room (VCR) that has been introduced to provide remote users with a virtual area from where they can control and monitor the instrumentation, and where they can collaborate with each others, even if located in different physical places. The third main component is the Execution Service, which provides a workflow engine able to handle BPEL workflows interacting both with the new features of GRIDCC/DORII and with traditional computational and storage Grid services.

The instrument used as a case study, to integrate sensors into the Grid, was a wireless sensor network (WSN) based on Imote2 modules. The main resources of an Imote2 are an Intel XScale 32-bit processor and a Radio Frequency (RF) unit based on the 802.15.4 standard. Each Imote2 can be connected to a sensor board that includes a 3D accelerometer and temperature, humidity and light sensors. After studying the alternatives for sensor operating system, it was selected TinyOS 1.x.

In the context of integrating WSNs into Grid civil protection applications, simulation is relevant due to the fact that:

1. it can replace real sensors with a simulated version, which makes it easier, faster and cheaper to test applications;
2. it makes it possible to scale a real network and test applications with a larger input data set;
3. it aids on the choice of the topology and localization of the network.

For these reasons, the University of Minho team evaluated a large set of network simulators, like GloMoSim, OPNET, OMNet, EmStar, SensorSim, ns-2, SENSE, ATEMU, TOSSIM, and DISENS. The comparison of simulators focused on the scalability, the popularity, the GUI, the platforms where it runs and the license of utilization. The selected simulator was TOSSIM, an accurate and scalable simulator of entire TinyOS applications.

8.2.2 System Architecture

The architecture of the system proposed to integrate an Imote2 based WSN instrument into the Grid is synthesized in figure 8-3. The main components of this architecture are:

1. the wireless sensor network itself and its base station;
2. the instrument access services host with the CJI, observations database, SOS, and other related services;
3. the host of the Instrument Element, which may contain several Instrument Managers and associated custom plugins to interface with CJI;
4. the VCR host that allows users to access the instrument.

Each remote Imote2 module, with associated sensors, will run a customized nesC application that monitors temperature, light, humidity and 3D acceleration. All Imote2 nodes of the network are connected to an Imote2 base-station through RF. The Imote2 base-station is serially (USB) connected to a laptop and the main task of the application running on it is to perform packet routing between the remote Imote2 and the laptop.

The laptop designated by Instrument Access Services host (IASH) will have a Custom Java Interface (CJI) for sensor operations and TCP/IP connection to the correspondent IM. The IASH also hosts the Observations Database and the OGC services (SOS, SPS, etc).

The remaining components of the architecture are the VCR and IE hosts, which in our implementation are installed in different desktops, both with Grid connection. There is a single IE for the instrument (WSN), but we can have several VCR users connecting to the IE at the same time. The implementation of the IM for Imote2 will include a custom plugin to interface with the CJI.

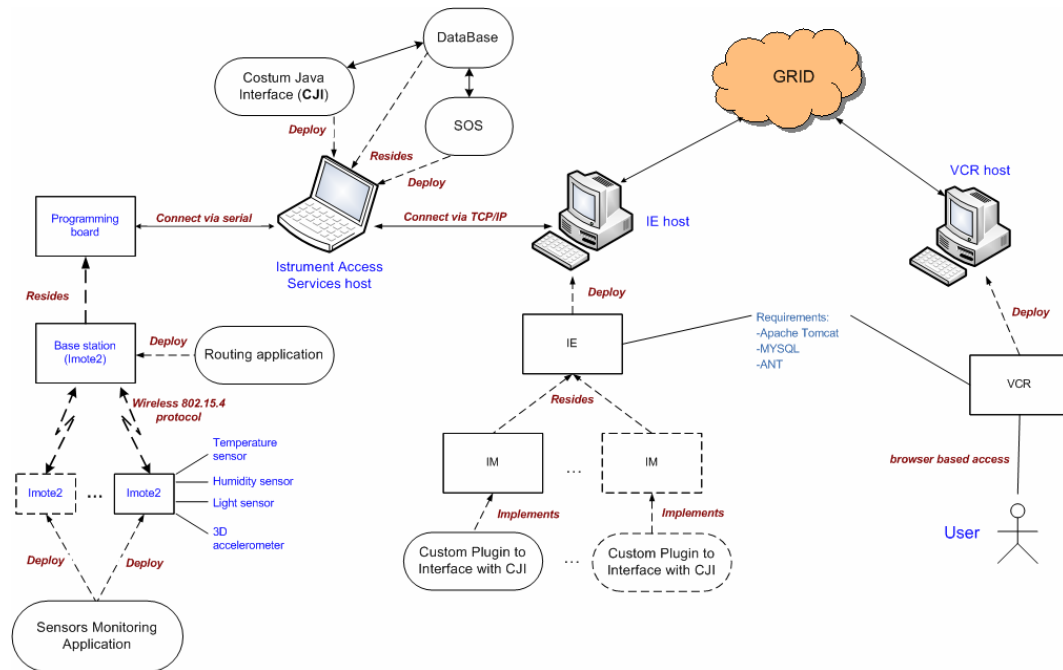


Figure 8-3: the architecture proposed to integrate a WSN instrument into the Grid

8.2.3 System Prototype

Our prototype system uses one IM for each Imote2, due to GRIDCC/DORII software constraints. Constraints were raised from some particularities of our WSN instrument, which were not foreseen during GRIDCC/DORII conception. For example, the particular way needed to manage time and duration of the jobs. Another particularity, that brings complexity to our case study, is the fact that we had to consider the real instrument a single Imote2, not the network of Imote2's. We could have virtualized the network itself as a single instrument, but it would result in a giant virtual instrument with a huge number of parameters, and the result would not be friendly at all.

We have developed the CJI to translate commands and data between the IM and the Imote2's and to manage the connection itself using TCP/IP. The CJI gained relevance in our prototype because we need to guarantee long-time jobs (sensor monitoring), even if the users are offline, and save the result's file, signalling the jobs that are complete. We decided to adopt OGC Sensor Observation Service (SOS) standard, to retrieve the results from a database and forward them to the user when he requests them. So we have the database populated by our CJI, according with the users requests, and extract those results using a SOS web service. The CJI also receives the user requests and processes them into single sensor readings, marking those readings with a tag that will identify the user and the job to which they belong.

The VCR provides the user with several commands to interact with sensors, and shows the state of some defined attributes and parameters.

As it was said before, the implementation of the IE contains an IM for each Imote2. The IM includes three essential components:

1. the cache used to store the most recent observation;
2. the SharedObject that provides the users with data from a previously submitted job, and
3. the implementation of the VCR commands (see figure 8-4).

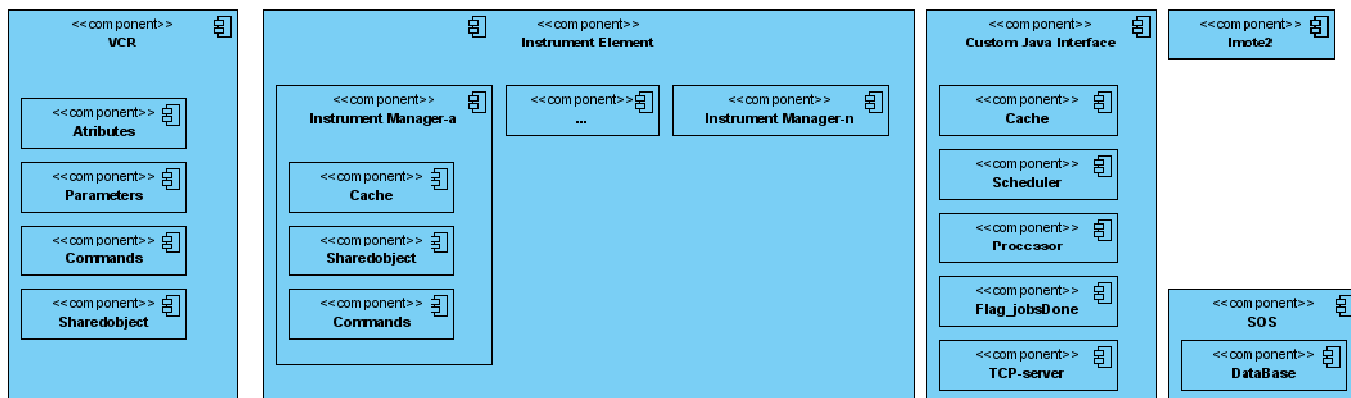


Figure 8-4: The software components of the architecture used to integrate a WSN into the Grid

8.2.4 Case Studies

The CJI was implemented with following components:

1. the cache, the unique element that really accesses the Imote2 nodes. The cache keeps the last value for each attribute, while it is valid. When the value is outdated, it asks a new observation from the Imote2;
2. the scheduler, that receives a periodic job (with user defined duration, periodicity, attributes), and transforms it in several single time scheduled observations;
3. the processor, a component that, at scheduled time, processes the requested observation using the values present in the cache and stores the data in the observations database;
4. the Flag_jobsDone, a flag used to inform the users that a certain job, previously requested, is complete, and the users can now get the correspondent observations;
5. the TCP-Server, which receives connections from the IM and send the requested data back to the IM.

The DataBase, where the observations are stored, is shared with the SOS implementation (see figure 8-4).

To overcome some limitations of GRICC/DORII instruments access, the University of Minho carried out a second case study: the implementation of a Web service, to access Imote2 sensors, based on the OGC standards.

In the OGC Sensor Web Enablement (SWE) activity, members of the OGC are defining, testing, and documenting a consistent framework of open standards for exploiting Web-connected sensors and sensor systems of any type. To perform this case study, we have analyzed two relevant implementations of OGC standards: one from NICTA, at Melbourne University, and the other from 52° North.

The basic standards for modeling and codification of the information involved in the access to sensors are Observation & Measurements (O&M) and SensorML. SensorML provides a functional model of the sensor system, rather than a detailed description of its hardware. O&M defines a framework and a conceptual model, where the encoding is formalized as an application schema. O&M binds a result to a feature of interest, upon which the observation was made using a procedure to determine the value of the result.

The standards that define essential services to access the sensors data are Sensor Observation Service (SOS) and Sensor Planning Service (SPS). The SOS provides an interface to make sensors and sensor data archives accessible via an interoperable Web based interface. The basic SOS allows the operations GetCapabilities (request a self-description of the service), DescribeSensor (request information about the sensor itself in a SensorML document) and GetObservation (request the sensor data, encoded in an O&M document). The SPS is an interface to task an asset or asset system. The SPS allows the operations GetCapabilities (retrieve metadata about the capabilities of an SPS), DescribeTasking (get the information that is required to prepare a tasking request), GetFeasibility (obtain feasibility details about a tasking

request), Submit (submit a tasking request) and DescribeResultAccess (retrieve information about where observed data can be access from).

In the second case study it was intended to read, control and analyzes data from a WSN into Grid civil protection applications, using the GRIDCC/DORII middleware but in conformity with the OGC standards. The figure 8-5 represents a simplified view of the proposed architecture, where one can see how to integrate sensors into the Grid, through the SOS standard. The used database schema was based on the 52° North SOS implementation. The core tables and definitions are:

- **feature_of_interest** the table that stores data about the features or feature of interest. It represents the identifiable object(s) on which the sensor systems are making observations
- **observation**, the table that aggregates the data of an observation event like time, procedure (sensor or group of sensors), the feature of interest and the observation value, which is stored in a separate table. An observation is an action with a result which has a value describing some phenomenon. An observation uses a procedure to determine the value of the result, which may involve a sensor or observer, analytical procedure, simulation or other numerical process
- **procedure**, it is the reference to one or more procedures, including sensor systems, instruments, simulators, etc, that supply observations in this offering
- **offering**, the table that stores each offering of the SOS. It is only used when the SOS is initialized to read in the offerings of the SOS and the phenomena which are related to each offering. An SOS organizes collections of related sensor system observations into Observation Offerings. Ex. Temperature in a station weather placed in a city
- **phenomenon**, it represents the phenomena (temperature, humidity, etc)

The SOS will use these tables to answer incoming requests, but it is not SOS responsibility to update the values in the tables. This task has to be fulfilled externally or using the CJI.

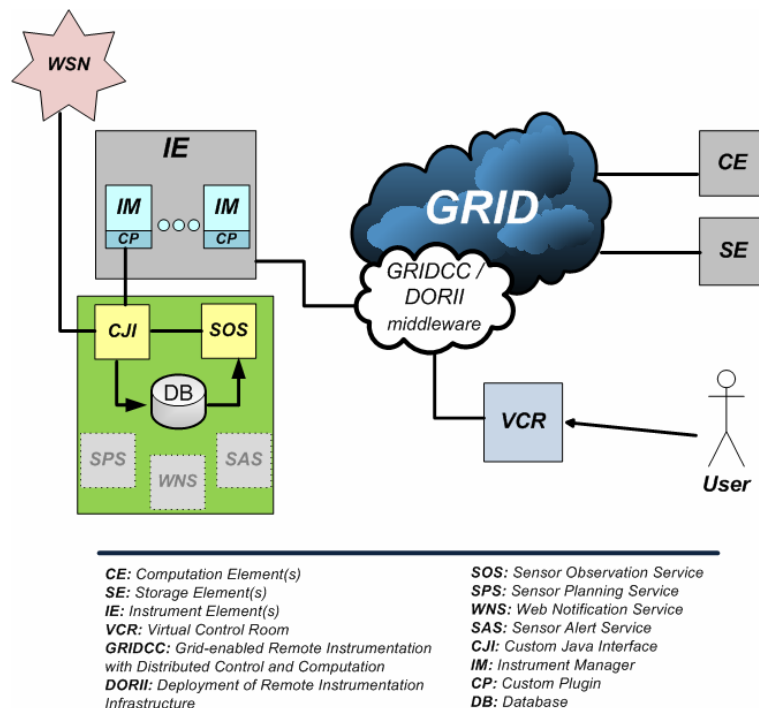


Figure 8-5: A simplified view of the architecture focused on the access to sensors through OGC standards (SOS)

8.2.5 Conclusions

A functional prototype to provide Grid civil protection applications with data from a remote wireless sensor network was developed. A first version of the prototype was based on the utilization of the GRIDCC/DORII middleware. This prototype was later improved with observations handling and services which are in conformity with the OGC SWE standards, O&M/SensorML and SOS respectively. SOS was implemented successfully, receiving XML requests and returning XML documents with the response.

A presentation about this work was accepted at IBERGRID'08 Conference:

- Esteves, J. Macedo, A. Serrano, V. Sá, L. Lopes, M. Caldas, “Enabling mote-based Environment Sensor Network for Civil Protection Grid Based Applications”, 2nd Iberian Grid Infrastructure Conference, Porto, May, 2008.

Full presentation: https://pop.cp.di.uminho.pt/cyclops/wp-content/uploads/2008/09/cyclops_ibergrid2008_sensors.pdf

8.3 PARTNERSHIP: ÉCOLE DE MINES – UNIVERSITY OF MINHO

Departing from the flash flood use-case analysis described in project's deliverable D9 [R14] and initial System Requirements in project's deliverable D11 [R15], École de Mines and University of Minho partnership decided to analyze in more depth the operational requirements imposed by CP applications on Grid platforms.

The objective of such collaboration was:

1. to identify any other requirements, such as. interoperability among heterogeneous, HPC requirements and interactivity
2. to achieve a solid base for collaboration between the two institutions and
3. to extend the Civil protection system analysis to all the project partners

In practice the collaboration was assumed as an agreement that determined a relation between École de Mines and the Departamento de Informática of University of Minho because Ricardo, the designated CYCLOPS collaborator from University of Minho, is also a Msc student. António Pina was the official from the sending institution and the tutor in charge of the placement and of monitoring the work was Pierre-Alain Ayrat with the help of Vincent Thierion, both from École de Mines.

The planned work would permit:

1. to consolidate partnership between the two CYCLOPS Partners
2. to run the model of prevision the flash flood ALHTAÏR on the Grid and
3. to work out on a model of parallelization of ALHTAÏR.

In what follow we report the analysis of Althair, a French civil protection flood modeling software, for future parallelization and porting to the Grid infrastructure. A parallel algorithm for the application is proposed, a model for its execution in a Grid environment making use of Open Web Services is designed.

8.3.1 Flash Flood

Flash food names a phenomena that occurs frequently in the South of France, in the Gard Region located between the Mediterranean and the first foothills of Central Massif, “Les Cévennes “ (Figure 8-6).

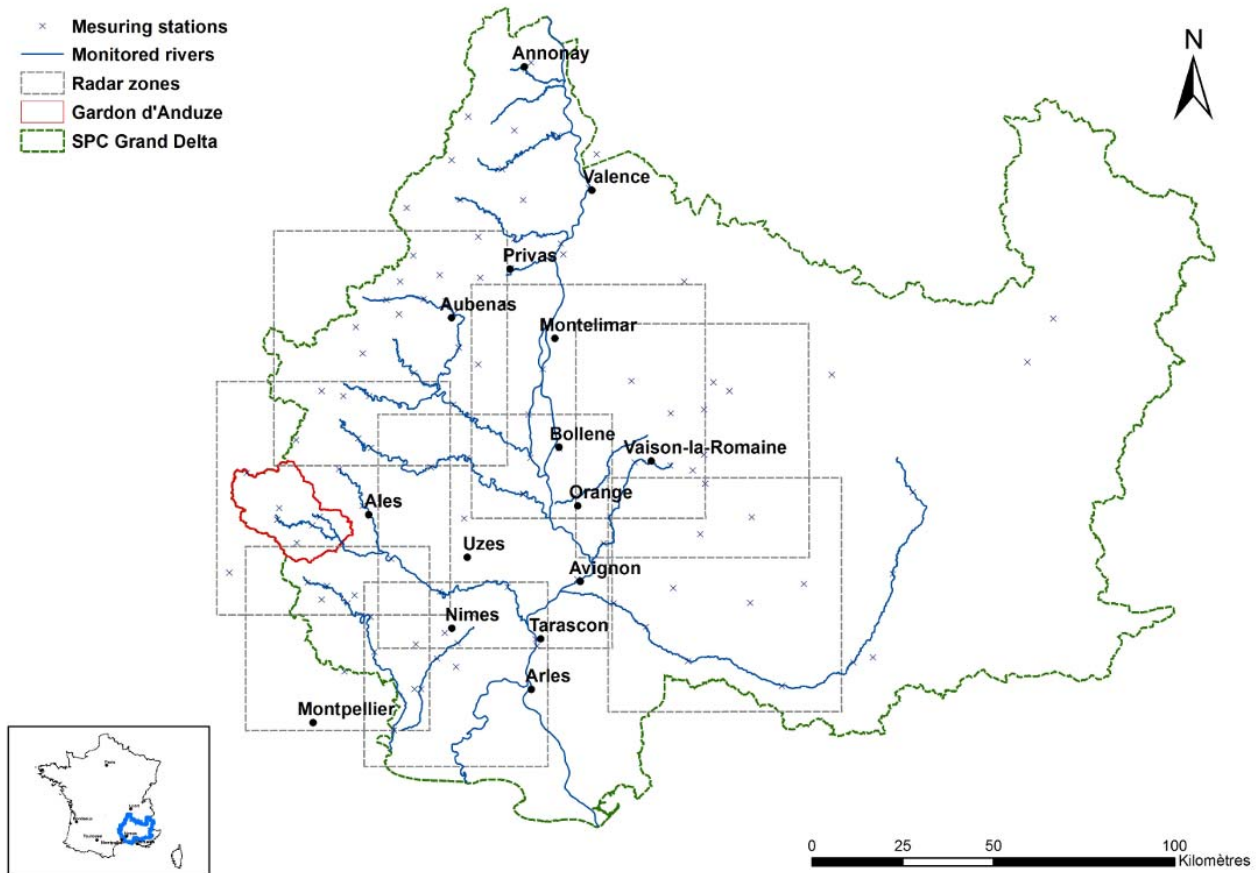


Figure 8-6: Grand Delta floods forecasting service competency territory

When southern wind pushes the moist warm air from the Mediterranean to a much colder coast due to the presence of a mountain massif, the conditions for the disaster are set up. The difference of the sea and the earth temperature makes the air cool down, and consequently the relative humidity increases. As a result we have high intensity rainfall event. This sequence of events is called “*épisode Cevenol*” (Figure 8-7), and generally occurs in the autumn. Associated to this meteorological event, a watershed rapid response causes a potential flash flood.

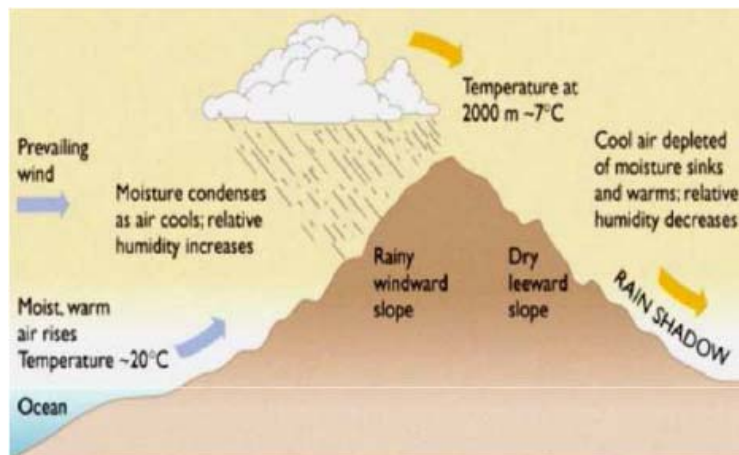


Figure 8-7: An “*Épisode Cevenol*”

Recently, in 2002, the damages caused by these foods made clear the need for improving real-time monitoring and food forecast. During the 8th and the 9th September, an extreme "épisode Cèvenol" occurred in the Gard Region. Its results were tragic, causing 22 victims and 1.2 billion €damages.

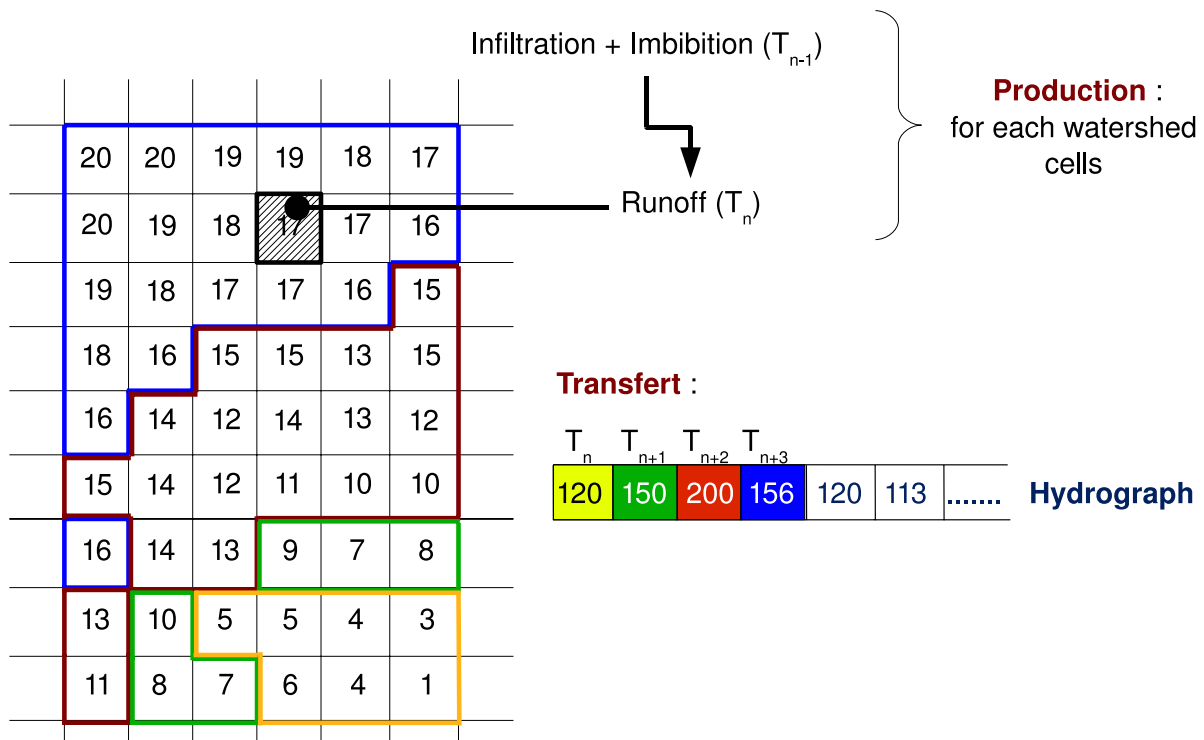
8.3.2 Alhtair

Alhtair (ALarm Hydrologique Territoriale Automatisée par l'indicateur de Risque) is an hydrological forecasting model (which corresponds to a distributed rainfall - runoff model) developed by the food warning service of the Gard Region, in the South of France.

The execution of Alhtair has two separate logic phases: the production phase and the propagation phase.

The production phase is as follows: based on the rainfall information and on the characteristics of the terrain, calculate the amount of water that infiltrates in the soil (flow), and the amount that stays in the surface (runoff).

Then, in the propagation phase, we calculate the volume of water per second that is expected to reach the outlet point for all given time intervals that correspond to the time between two consecutive rainfall reports. Figure 8-8 shows an example of a transfer matrix, used both in the production and propagation phases.



$$\sum \text{waterflow } T_n + \sum \text{waterflow } T_{n+1} + \sum \text{waterflow } T_{n+2} + \dots = \text{Simulated Hydrograph}$$

Figure 8-8: Synthesis on the Alhtair propagation module

8.3.3 A proposal for the parallelization of Alhtair

Currently, processing a rainfall report takes approximately 12 seconds, depending on the machine used to run it. If we add to this the (real) possibility of running the simulation with rainfall reports plus a set of forecast reports, then the

simulation time can increase dramatically. As an example, if after receiving a rainfall report we want to add to the result a set of 60 forecast reports corresponding to a forecast of 5 hours, then the simulation would take an average of 12 minutes. This makes the need for reduction of the simulation time very clear. The idea is to make a simulation in the time interval between the arrival of two consecutive reports, that is, 5 minutes. This simulation should be as rich as possible, taking into account as much forecast reports as possible.

The most forecast reports are used, the more reliable the final result is. Figure 8-9 shows this characteristic of the model.

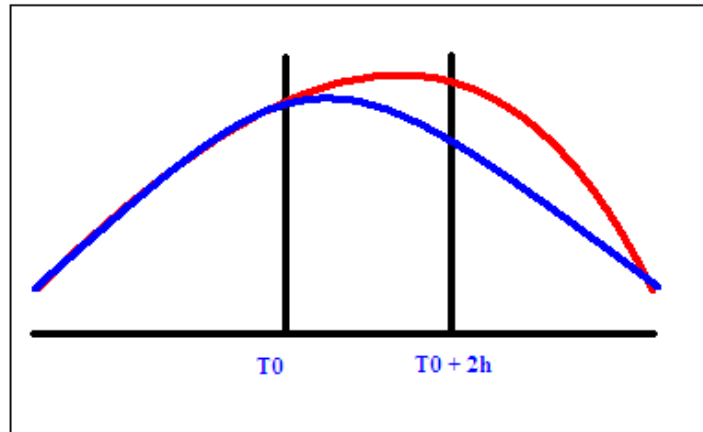


Figure 8-9: Reliability of the results of the simulation

In blue we can see the expected hydrograph as a result of the execution without forecast reports, and in red the expected result with 2 hours of forecasts reports included. The results for both executions are the same until T_0 . This is a consequence of using the same rainfall reports. From T_0 on, the blue hydrograph gives a constant decreasing of the flow, due to the absence of rainfall from that point on. On the other hand, the red graphic keeps increasing until $T_0 + 2h$, due to the rainfall prevision registered in the forecast reports for that period of time. From that point on, the hydrograph decreases due to the absence of rainfall forecast reports.

The number inside each cell of figure 8-8 represents the time that the water flow takes to reach the outlet point (transfer time). In this case, the outlet is located in the lower right corner, as we can see by the value of 1 present in this cell. The cells are grouped by transfer time. On the top right of the figure 8-8, we can see an example of how the water flow is calculated, as described before. The final result of the simulation is achieved by summing the total runoff in the watershed for each time interval T_n .

Finally, this first level of parallelization concerns the core algorithm of Alhtair. Indeed, the watershed can be split in regular group of independent cells on which raw rainfall is transformed in infiltration and water flow for each cell. The propagation module transfers water to the outlet of the watershed and creates the hydrograph of river runoff. Thus, a group of dependent jobs have to manage the two modules to create the corresponding hydrograph, but parallel processes can be also used to simulate each independent watershed with different rainfall conditions.

8.3.4 Flood forecasting process through Grid Enabled OWS

The parallelization of the production stage could effectively improve hydrological simulation time-of-response. But to perform this operational objective, a efficient jobs and data management has been necessary. Indeed, about 30 watersheds spread out the Grand Delta territory have to be simulated in a critical phase of 5 minutes. For each watershed, expert can request several simulations based on given rainfall variations (a rainfall intensity variation of 50% occurs 20 additional jobs whose the duration depends on the requested forecast delay). Thus, in an extreme hydrological situation implying the whole of the Grand Delta territory, the system could have to execute more than 600 jobs in a short-term delay. The development of a global infrastructure based on Grid enables OWS technology seemed mandatory to the successful of this operational porting.

Thus, experience and experimentations handled on the RISICO application porting already presented in previous section has been adapted to the specific requirements of flash floods forecasting system. This represents a important collaboration between IMAA and EMA-Armines partners. Web services potentialities have been demonstrated for the forests fires use-case in designing specific geospatial web services (OWS) such as WCS and WPS. IMAA experience has enabled to adapt these web services to the Alhtair new system.

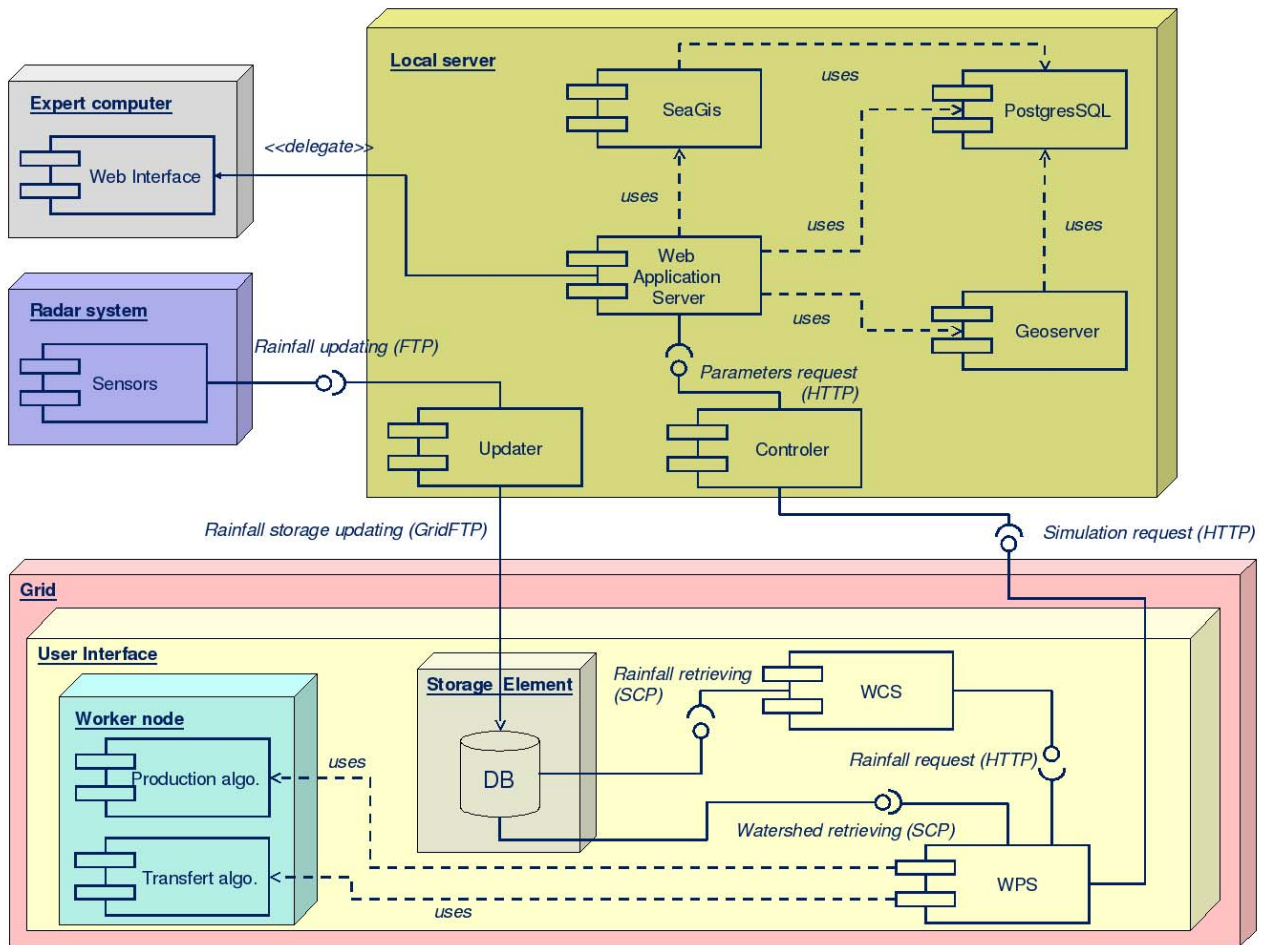


Figure 8-10: UML component diagram of grid enabled Alhtair simulation

The real simulation using only 5 minutes rainfall data is launched on the local system and provided watershed conditions to the next simulation and to the potential forecasting processes executed on the grid. Two local modules are integrated and managed by TOMCAT server:

- the Updater stores rainfall data in multi-dimensional file format (NetCDF) and shows rainfall situation on the GUI
- the Controller which manages the preparation of simulations execution in interpreting user requests (GUI) and in transforming them in XML request for Web Processing Service (WPS) functioning. In post simulation stage, the Controller interprets WPS response in reading NCML syntax corresponding to a numerical array and creates human-readable hydrographs.

These modules are developed in python and use NetCDF and Numpy libraries.

Once real-time simulations are finished, original rainfall forecasts data provided by radar system are stored in a SE, as well as watershed characteristics and previous watershed hydrological conditions.

The developed WCS and WPS facilitate the management of these data and these jobs. The WCS subsets and prepares watershed, rainfall and hydrological conditions data to enable multiple simulations processes. Two WPS have been created for each module of Alhtair, one dedicated to the production stage which controls data preparation provided by WCS and core production algorithm and its outputs, another for transfer module also using WCS services.

The core algorithm needs two specific libraries (Numpy and NetCDF) set up in each site or included in the InputSandbox of every job. As already mentioned, these data transfer and job execution operations are taken in charge by the Wrapper through the WPS.

8.3.5 Conclusions

The objectives of this work were fully achieved. The parallel version of Alhtair has the potential for decreasing the execution time of the application and also to provide the decision makers with more reliable information in real time. This is a consequence of the use of a bigger number of reports, including forecast reports.

Alhtair in progress developments should confirm, as RISICO experimentations, the effectiveness of Grid Enabled OWS technology to support intensive and time critical applications. From a technical point of view, use of geospatial web services facilitates interactions between local systems represented by Civil Protection actors (end users) and Grid technology infrastructure and seems to provide a standard and flexible layer for many different Civil Protection applications.

A report about this activity is available at: <https://pop.cp.di.uminho.pt/cyclops/wp-content/uploads/2008/10/alhtair.pdf>

8.4 COLLABORATION DORII – UNIVERSITY OF MINHO

During CYCLOPS 2^a Training Workshop Andrea Del Linz, from Sincrotrone Trieste S.C.p.A, one of the partners of DORII project, presented to the audience the motivation, the main objectives and the target applications of the DORII project. The session also included detailed introduction to IE (Instrument Element) and VCR (Virtual Control Room) - concepts developed by GRIDCC project in the context of the Deployment of Remote Instrumentation Infrastructure into the Grid - and a Demo used as proof of concepts.

Following this first contact with DORII project, in order to get a closer collaboration in the related areas of research and development, we decided to accept Dr. Roberto Roberto Pugliese kind invitation to visit them at Elettra Laboratories, in Trieste. In-house research and development is pursued at the Elettra through qualified projects supported under the EU Framework programmes as is the case of DORII.

During the visit of Vítor Sá, collaborator of CYCLOPS at University of Minho, which is also an MSc student, to the Elettra laboratories, António Pina was the official from CYCLOPS and António Esteves the scientific advisor from Department of Informática from Universidade do Minho. At Elettra laboratory Andrea Del Linz was the tutor in charge of the placement and of monitoring, although the work has been characterized by a large autonomy from the staff of DORII.

8.4.1 Problems addressed and results

The planned work would permit to study and adapt the software developed by GRIDCC / DORII project for use in monitoring and managing Wireless Sensor Networks in the context of the CYCLOPS project. More specifically, as a case study, we intended to adapt the existent software to be able to control a network of wireless sensors using IMOTE2 nodes.

In what follows the work that has been done is reported by a sequence of items that highlight the topics discussed and the correspondent results.

- Presentation and installation of the new software version "Virtual Control Room" (VCR) and "Instrument Element" (IE):

the DORII team revealed the new features implemented in the latest version of the software, and has given us access to the program modules to use in our work. They also made an installation and configuration of the system answering to our questions, and confirming procedures

- Participation in the definition of features to implement (or change) in future versions of the software DORII:

throughout the entire visit we suggested features that our previous work had proved to be useful or even essential for a possible and hopefully effective use of software DORII for the management of a wireless sensors network under CYCLOPS. Contributions were welcome and several topics related to the development of software, were discussed. When there was already an implemented solution for the desired features we also have been informed of the best way to use and configure the software for the particular needs of our work.

- Virtualization of the sensor IMote2:

the parameters and attributes that should be implemented, in order to meet the requirements of our work in the CYCLOPS, were discussed, and the DORII team accepted the need to improve certain features in a forthcoming version of the software. Those features are necessary for properly (and practically) virtualizing sensors in real situations, because none of the previous case studies focused on networks of sensors, so that the particularities and requirements of those instruments we were not familiar to them.

- Error management in the implementation of sensor networks:

we discussed the best way to address the errors which may arise during the operation of a wireless sensor network, and the advantages or disadvantages of defining a partial or total error state for the instrument sensor/network-of-sensors. We concluded that the software DORII needed to be refined in this respect, because our instrument had an unexpected complexity to the DORII team, and methods of thinner control proved now to be very useful. I was told it would study the feasibility of implementing those features in a future version of the software.

- Study of alternatives, and possible implementations of the "Instrument Manager" (IM) for sensors / networks of sensors:

again, the complexity brought by our instrument was new to the DORII team. And even though some needed features have already been outlined, they had not yet been implemented. The issues we raised show them the practical usefulness, so they decided that its implementation would no more be postponed.

- Study the problem of control and policies on use of the VCR:

the way the VCR was implemented, their policies of use and methods of control started from very simple assumptions, which were driven by the complexity of software implementation combined with the restrictions of the GRID environment, and ease of use intended for users. The problems we presented were already familiar to them but they had no immediate solution for them. So, it was clear that we should live with these limitations and not losing time trying to resolve them.

- Definition of commands needed to implement the management model you want:

again the deep knowledge - own by the team that implemented the software - was of major importance, because it set what was not possible and the best way to use the features already available to sensors monitoring and schedule observations. Again, we come to the conclusion that the software still had some weaknesses to be used with this kind of instrument in particular (wireless sensor networks as part of civil protection), and the team assumed the interest to overcome some of these weaknesses in short term (a few were already being worked on that date), whereas others could not realistically be corrected, unless maybe in the long term.

- Study the problem of monitoring and visualization of geo-referenced data using the Google-Maps API (available in GRIDCC):

nowadays, the monitoring is limited, and we hope there is a version much more functional and dynamic in a future version of the software.

This visit influenced enormously the work University of Minho presented in the above sub-section 8.2 Integration of Sensors.

8.5 COLLABORATION NGI / EELA-2 / INT.EU.GRID – UNIVERSITY OF MINHO

As a consequence of the discussion that involved all the participants of CYCLOPS WP4 Technical Meeting, University of Minho considered the possibility to go a step further in the need of:

- a) to clarify the advantages in using the Grid environment for CP application,
- b) to analyse general approaches to the parallelization of computing tasks for CP applications and
- c) to testing collaborative and user friendly frameworks for running interactive applications

The purpose may also be seen as a response to the First Review Recommendations that state:

- 1) The project has to establish stronger collaborations with “ other e-Infrastructures ... other related projects and initiatives ... institutions and organizations in Europe ... and worldwide ... engage actively in collaboration with them and promote the CYCLOPS results by as many projects as possible..”
- 2) “the consortium should analyze in more depth the operational requirements imposed by CP applications on Grid platforms and identify any other requirements (e.g. interoperability among heterogeneous) to communicate to the Grid community.

8.5.1 Portuguese NGI

At INGRID'06 meeting - the first public preparation meeting of the Portuguese Grid Initiative at Universidade of Minho (Braga) – Paolo Mazzetti presented the CYCLOPS project.

This event may be seen as the starting point of University of Minho participation in CYCLOPS since it leverages the appearance of Cross-Fire, an R&D proposal submitted to a call for GRID projects under the Portuguese National Grid Initiative.

CROSS-Fire: Collaborative Resources Online to Support Simulations on Forest Fires, is being funded since September 2007 by FCT: a Governmental institution that funds and evaluates R&D Units in diverse scientific areas (see <https://www.fct.mctes.pt/>).

The project aims to complement national and EU projects, pursuing R&D in computer and computational sciences, demonstrating its potential through a CP activity application and the deploying of interoperable grid applications and services, among several independent organizations.

Cross-Fire includes FireStation, a CP application, created at ADAI-University of Coimbra as the result of the research on forest fires, started in 1986 and is being kept continuously with the support of several national and EU funded projects. It is an integrated application composed by several modules: the simulation of fire growth over complex topography, the calculation of the wind field as well as one module for the computation of the Fire Weather Index based on the Canadian System.

The porting of FireStation to grid could benefit from the recent achievements of CYCLOPS since it has Functional Requirements equivalent to those of CYCLOPS user-cases, such as the need for Geo Information access, Data Policy Support, or Authentication and Authorization. On the other hand the grid community and EU Civil Protection may also benefit from the results that may be achieved by Cross-Fire project.

Collaboration started with the creation and maintenance of a supporting Grid infrastructure supported by the resources of CROSS_FIRE project since, at that moment, CYCLOPS had no Grid resources in Portugal. The development of the EGEE Roll which has been later used as a Framework to Fully-automated site Deployment & Management is a direct consequence of that collaboration.

At the application level we decided to establish a common road map to pursue recommendation 2 (see above) taking into account the following issues for FireStation on the grid:

- high parallel computing and grid storage,
- interoperability problems, spatial data management,
- near-real-time requirements,
- QoS data policy and security (resource prioritization/reservation),

- visualization and geo-spatial information sharing.

The establishing of contacts with the scientific community that participates in Cross-Fire, such as ADAI-University of Coimbra, was of great importance to CYCLOPS dissemination in Portugal. ADAI is a non-profit association aiming the implementation and development of knowledge in creation and management of scientific and technological infrastructures, in particular the Forest Fire Research Laboratory. In particular Xavier Viegas, its director, participated in two CYCLOPS sessions at Ibergrid'08:

1. Project CYCLOPS Panel European Civil Protection Infrastructure.
2. Decision Support Tools in Forest Fires Management (see http://www.ibergrid.eu/2008/presentations/Dia%2014/6c_paper2.pdf)

8.5.2 Int.EU.Grid

Collaboration with the Int.EU.Grid (The Interactive European Grid) project is a direct consequence of the contacts established with LIP (Laboratory of Instrumentation and Experimental Particles Physics) during the installation of Uminho-CP site.

This project infra-structure complements EGEE by providing a framework specifically oriented to interactive demanding applications, including MPI usage, powerful visualization, and a collaborative environment.

LIP participates as partner both of EGEE and Interactive European Grid.

At the moment, the exploitation of the facilities and the results achieved by Int.EU.Grid project, both at the grid middleware and infrastructure levels, seems to be the solution to several of the issues posed by the gridification of FireStation. Just as an example, gLite 3.1 includes several updates, such as the support to MPI resulting from its project achievements.

Immediate results of the contacts with LIP resulted in:

- 1) availability to support CYCLOPS VO for the execution of CYCLOPS user-cases
- 2) access to a RAS server to evaluate the interactive environment facilities provided by MigrationDesktop and
- 3) support to the installation of a new site, belonging to Int.EU.Grid grid infrastructure: GRID-PROCIV (pop.grid.prociv.pt), to evaluate the interoperability with EGEE and interactive grid added value, in particular in what concerns steering, graphical display and visualization needs.

8.5.3 EELA-2

EELA is an EU funded project (under FP6) aimed to build a bridge between consolidated e-Infrastructure initiatives in Europe and emerging ones in Latin America. It also aims create a collaboration network to deploy a large portfolio of scientific applications on a well supported Pilot Test-bed. EELA-2 is a second edition of the same project (under FP7) formed by 54 members that foster to provide an empowered Grid Facility with versatile services fulfilling application requirements.

Cooperation with the EELA-2 is a natural consequence of University of Minho participation in EELA consortium, integrated in the Portuguese JRU. Cross-Fire (FireStation) is the sole Civil Protection application in a universe of 50 other applications being supported by EELA-2 project.

This context favours, enormously, CYCLOPS results dissemination to communities outside the EU. In addition, collaboration with EELA also resulted in the following full supported training activities, being pursued by members of the CYCLOPS team:

- 1) Joint EELA-2/EGEE-III Tutorial for Trainers - to prepare people to assume training roles in EGEE tutorials, specially, but not limited to, in their countries, by bringing to participants the new services of gLite 3.1 and the additional services of JRA1 already available.

(<http://indico.eu-eela.eu/conferenceOtherViews.py?view=standard&confId=125>)

- 2) The First EELA-2 Grid School (E2GRIS-1) - to create the necessary environment in Latin America for the porting of new applications to the EELA-2 Grid Infrastructure. To fulfil this goal, the E2GRIS-1 programme, based both on lectures and hands-on practices, will cover in-depth the main topics of Grid technology as well as the EELA-2 middleware technicalities. (<http://indico.eu-eela.eu/conferenceDisplay.py?confId=159>).

In particular, the Grid school, planned from 2nd to 15th November 2008, is of extreme importance since it allows using FireStation as another CP gridification use-case, to evaluate EGEE support to data storage and metadata manipulation of large data set. It will also allow evaluating MPI program design and execution applications, by means of AMGA.

GENIUS, a powerful Grid Portal that allows scientists to exploit Grid resources only using a conventional Web browser, it will also be used as a “natural” GUI for the gLite middleware, that already has many functions implemented (i.e. authentication) and allows to faster development due to standardization.