Project no. 034567

Grid4All

Specific Targeted Research Project (STREP)

Thematic Priority 2: Information Society Technologies

D4.5 Integrated Prototypes of applications

Due date of deliverable: 01-06-2009

Actual submission date: 11-07-2009

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
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)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Contents

Abbreviations used in this document

Abbreviation/Acronym	Description
API	Application Programming Interface
CFS	Collaborative File Sharing
CNSE	Collaborative Network Simulation Environment
DCMS	Distributed Component Management System
DHT	Distributed Hash Table
P2P	Peer-to-Peer
SIS	Semantic Information Service
VO	Virtual Organization
VOFS	Virtual Organization File System
VOMS	Virtual Organization Membership Service
XML	eXtensible Markup Language

Grid4All list of participants

Role	Participant N°	Participant name	Participant short name	Country
CO	1	France Telecom	FT	FR
CR	2	Institut National de Recherche en Informatique en Automatique	INRIA	FR
CR	3	The Royal Institute of technology	KTH	SWE
CR	4	Swedish Institute of Computer Science	SICS	SWE
CR	5	Institute of Communication and Computer Systems	ICCS	GR
CR	6	University of Piraeus Research Center	UPRC	GR
CR	7	Universitat Politècnica de Catalunya	UPC	ES
CR	8	ANTARES Produccion & Distribution S.L.	ANTARES	ES

1 Executive Summary

This document is part of the research project Grid4All (IST-FP6-034567). The nature of D4.5 is "Report, Prototype". The document describes the final integrated prototypes of the following applications developed in work package 4: CFS. CNSE, eMeeting, Sakura (Shared Calendar). The prototype applications are available at the public web page: http://www.grid4all.eu.

This document has to report the experience, what has been gained, obstacles and lessons learned during the development of the applications using the Grid4All infrastructure.

2 Introduction

This document is part of the research project Grid4All (IST-FP6-034567). The nature of D4.5 is "Report, Prototype". The document describes the final integrated prototypes of the following applications developed in WP4: CFS. CNSE, eMeeting, Sakura. The prototype applications are available at the public web page: http://www.grid4all.eu.

This document has to report the experience, what has been gained, obstacles and lessons learned during the development of the applications using the Grid4All infrastructure.

Details on the features of each application, the social environment where the application could give support, the technical environment that is required or where it works best, results and lessons learned and user manuals are described in more detail in "D4.8 User Manual" (M36). This document updates D4.3 "Prototype API usage by Grid4All based applications" and D4.4 "Initial prototypes of all applications" (M24) on the status of the applications at the end of the project. Aspects related to traces and evaluation are explained in more detail in D5.3 "Final integrated proof-of-concept implementation and evaluation report" (M36).

For each of the application, we describe:

- The features: what actions it supports or how it enables users, under which conditions
- Rationale for choosing each application and how it relates to the vision of the democratic Grid
- The environment in which it should work, and the environment in which it does work (this is further explained in more detail in D5.3).
- Discussion on lessons learned, summarized also in D4.8.

3 Collaborative File Sharing (CFS)

The purpose of the Collaborative File Sharing (CFS) application is supporting users when they collaborate, interact and share information in an asynchronous manner (they may not work at the same time). CFS works with workspaces: files and discussion forums shared by a group of participants. Workspaces may contain forums, files and other workspaces thus allowing a hierarchical organization if needed. Users may access an existing item (a file, forum or folder) or create a new one. Access to items may be restricted by the creator (each object has a list of access rights). In addition, a notion of role is supported (to label a set of access rights). All participants in a workspace have roles for every object in the workspace that determine what the user can do with each item. This role may be changed at any time by the creator or the item administrators. Interaction and collaboration between participants may be either through messages posted in a forum or through file upload, creating new versions of them.

The final prototype provides the following features that can be performed concurrently as it uses Telex to automatically resolve conflicts among concurrent actions:

- Management of a shared directory tree (workspace) structure: create, delete, move, change, list;

- Management of files: create file, create version version, delete, move, rename, list versions, read version;

- Management of forums (messages): post message, reply, view, modify, delete.

Therefore CFS relates to the vision of a democratic Grid (facilitating usage and supporting collaboration and data sharing) as it provides sharing of files, folders, and forums in a conflict-free manner where all participants always have the same view, by using Telex and VOFS and indirectly other Grid4All middleware elements.

The development ended by October 2008. Difficulties with the use of Telex (stability of Telex, due to the concurrent development of Telex and VOFS) limited the progress of development and the disconnected mode was not implemented in CFS as it was not supported by Telex at that time. The application has been evaluated in an educational environment after the development ended so the feedback obtained from the evaluation cannot be introduced in a revised software release. The lessons learned are presented together with the evaluation report in more detail in D5.3, but the main conclusion is that automatic conflict resolution is not that visible or necessary in face-to-face collaborative learning sessions where CFS has been evaluated (same time, same place, small groups). Conflicts does not happen so frequently are if so, they can be easily solved by social interaction. The automatic resolution may also be confusing to the participants as the current implementation does not ask or notify the user, it simply takes a default action. CFS might be more useful in a less coupled scenario (different time, different place) where direct interaction among participants is less feasible, and even more in scenarios where participants work in disconnected mode. The notification of conflict resolution is another area of further improvement. In addition, given that the application is in its first prototype, there are a few minor bugs, usability issues, and restrictions on the range of configurations (software requirements or dependencies) where it works.

Deliverable D4.3 and D4.4 already described the application architecture, operation and dependencies. D4.8 provides more details on the final prototype of CFS and its user manual. D5.3 presents the evaluation report.

4 eMeeting

eMeeting is an online synchronous collaborative tool that allows to share not only audio and video but documents (like video, pictures or text). It can be seen as a perfect companion for the rest of the Grid4All applications as it contributes a multi-channel real-time interactive session among the participants of a collaboration, giving the possibility to share their results, process or simply support decision taking on a project without the delays of asynchronous communication contributing with a real time collaboration tool to the Democratic Grid.

Participants are able to see and hear the tutor, control who can present material, carry out speeches, or solve specific doubts of a small group.

The attendees participate actively in the session, either on their own initiative, by asking to speak or at the prompting of the tutor, who can post them questions or request that they take control of any of the documents being shown to them. It is even possible to invite them to discus amongst themselves.

There is a “document area” where documents uploaded by the tutor can be downloaded by any participant.

The main eMeeting features are:

- **Session creation**: Session scheduling through the Grid4All Shared Calendar.
- **User Authentication**: User authentication to control the session access through the Grid4All VOMS.
- **Slideshow control**: Multimedia and text files storage through the Grid4All VOFS.
- **Audio/voice videoconference**: A centralized system which has not been modified in Gird4All as it requires real-time media flows.
- **Online chat**: a chat tool for keep a text conversation among users.

The interface is fully customizable and configurable, and the application can be accessed from anywhere, without it being necessary to install a program beforehand (just the Flash plug-in).

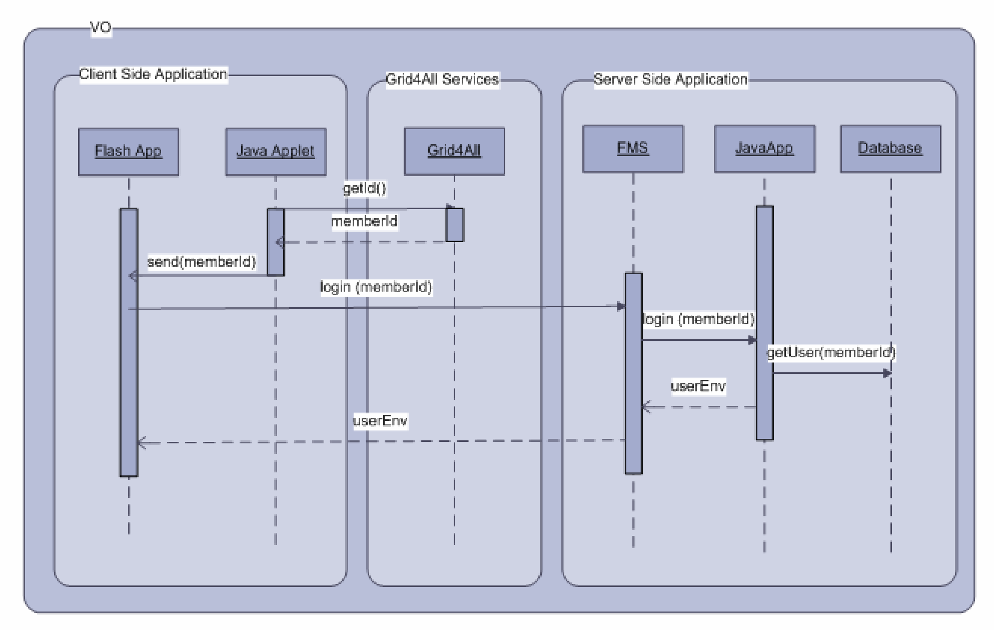
The eMeeting core is divided in two parts, the administrative interface and the end-user front-end. The administrative interface allows to control the basic operations of an eMeeting Session and thhe operations are performed through several web services.

The front-end application is Flash based and it uses some Action Message Format services from Flash Media Server to get the session data. It also connects with a Java applet that allows to get some info from the local system (as VOFS and VOMS).

This web service based administrative interface allows to create several satellite applications to administrate multiple instances of the client application. Each instance of the administrative application can administer (create, modify and delete) users, sessions and documents.

eMeeting provides Grid4All members the ability to communicate in synchronous sessions. eMeeting had the challenge of linking a centralized web conferencing application to a decentralized desktop application implied major changes to accomplish this objective.

To conclude, eMeeting shows that an already developed centralized web application can be partially integrated with the Grid4All vision. In this way it opens doors for the current international market of Rich Internet Applications to the Grid4All vision and results and they could be integrated as follows:



Even when video and audio conferencing is fully working, the technology used doesn't allow to decentralize the complete application, so one of the future steps would be to check any other decentralized audio/video system.

Deliverable D4.3 and D4.4 describe the application architecture, operation and environmental conditions. D4.8 provides more details on the final prototype of eMeeting and its user manual.

5 Collaborative Network Simulation Environment (CNSE)

CNSE is a grid service-based collaborative environment for network educational purpose. CNSE is implemented following the grid computing paradigm, and it supports the well-known open source *ns-2* network simulator, as well as the *nam* and *xgraph* for graphical representations. CNSE has been designed to overcome the limitations of *ns-2* related to the access to remote computational and storage resources to run multiple simulations. In addition, it supports collaborative learning scenarios, providing a number of features that are not currently offered either by *ns-2*, such as collaborative services for visualization and group awareness, and access to computational and storage resources required to carry out multiple simulations (i.e parameter-sweep simulations) in a reasonable period of time. Its current use is to support laboratory assignments in Computer Network courses in the University of Valladolid (Spain).

There are two prototypes of the CNSE regarding its functionality. Both designs have been developed with the grid service technology, splitting their functionality in a set of services. Each of these services can be offered by one or more organizations participating in the grid by running them using their own local resources. The CNSE is composed by a set of services and clients as it is shown in Figure 1. The clients can be student-oriented (*Participation Client*) or educator-oriented (*Management Client*). The services allow the execution of multiple simulations and visualizations of results as well as support collaborative scenarios. Besides, an input zip file must be provided to the prototype to define the number of simulations to be run, the input parameters for each of them, the expected output files, and the input *ns-2* file. A short description of the services is provided next:

- *Shared Repository Service*. It stores input files with the scripts and configurations for the simulations and output files with the results. This service can use resources distributed in the grid increasing the amount of data that can be stored.
- *Simulation Service*. It carries out the simulations of the network scenarios with *ns-2* according to the input file. CNSE can use multiple instances of this service in several organizations of the grid, opening up the possibility of running large number of simulations in a reasonable time.
- *Analysis Service*. It makes a statistical analysis of the data output files. This analysis is not limited to the local resources, hence it can use computational resources in the grid for the mathematical operations.
- *Index Service*. It provides service registering, publishing and discovering by the organizations participating in the grid.
- *Shared Visualization Service*. It generates graphical representations from the output simulation files with *nam* and *xgraph*. It allows collaborative visualizations among several members and groups in the grid.
- *Awareness Service*. It saves a user list with the current learners online, and their main actions.
- *Group Context Service*. It stores information and references about learners in each group that can be useful for the other services as a shared context of each group..

- *Credential Repository Service*. It stores user credentials for an authorized access to the application into the grid.

The first CNSE prototype just implements the following services *Group Context Service*, *Awareness Service*, *Shared Repository Service*, *Shared Visualization Service*, and *Simulation Service*. Unfortunately it can not run parameter-sweep simulations. In fact every simulation must be launch individually through the simulation service. So a second prototype has been developed to allow parameter-sweep simulations.

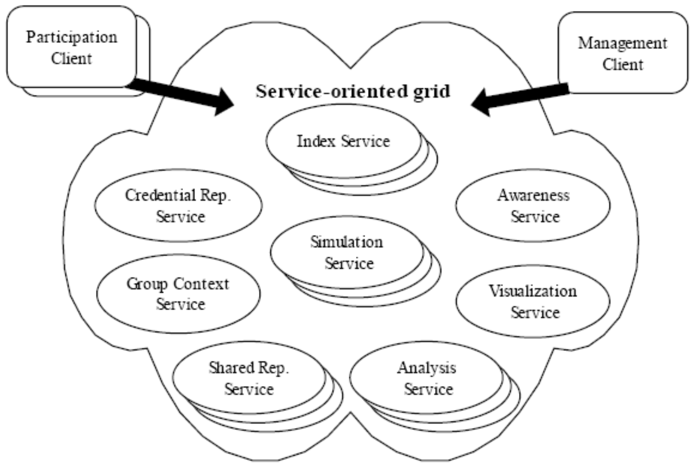


Figure 1. CNSE. Service-based original design

The second prototype splits the functionality of the *Simulation Service* in three different services: *Broker Service*, *Scheduler Service* and the *Simulation Service* itself. Figure 2 shows the service composition and their flow to run a parameter-sweep simulation and represent its visualization afterwards. In this prototype a simulation package is needed as an input from the *Participation Client*. This package contains not only the input script, but also an XML manifest with the parameter-sweep simulations to be executed and the output files from the ones that have been already run.

According to the Figure 2 the process starts with the *Participation Client* loading an input package with some parameter-sweep simulations already defined. The user (student or educator) can add, edit or delete a single or parameter-sweep simulation by assigning parameter values. Once selected, a single or parameter-sweep simulation will be send to the *Broker Service* (1). Next, the *Broker Service* queries the *Index Service* to get a relationship of the available *Simulation Services* with some performance measures (such as the time to run a short test simulation) (2 - 3). Then, the *Broker Service* invokes the *Scheduler Service* with a relationship of the simulations to be done (*tasks*) and the number of services, splitting *tasks* into *jobs* for each service (4 - 5). Every single job is sent to the *Simulation Service* to perform the simulation individually (6), and the results are merged and returned back to the *Broker Service* and consequently to the *Participation Client* (7-8). Finally, the student decides to visualize the results, so it queries the *Index Service* to get some available *Visualization Service* that represents the results with *nam* (as an animation with packages running from one node to another, being discard or not)or *xgraph* (as a plot with an input parameter for the x axis and an output value for the y axis).

The second prototype of the CNSE does not include collaborative features yet. However, it is intended to integrate the *Shared Repository Service*, the *Group Context Service*, and the *Awareness Service*.

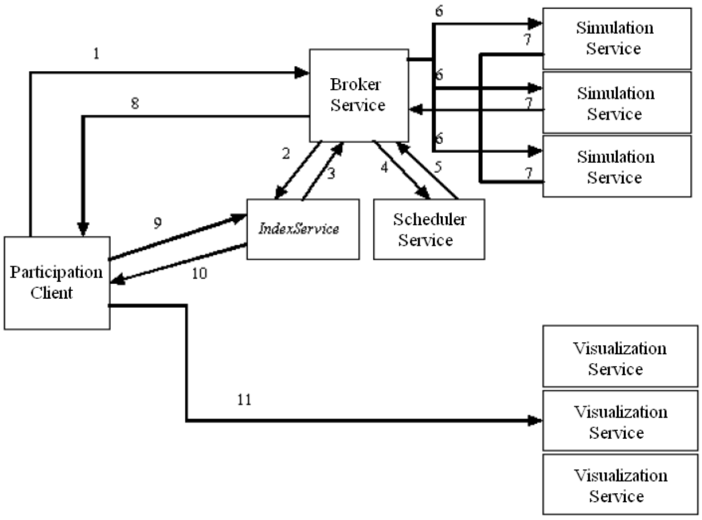


Figure 2: Steps to run a simulation and launch a visualization of the results in the Parameter-Sweep CNSE.

CNSE is involved as part of the Grid4All project to envision democratic Grid in the field of Education and Collaborative Learning. Indeed, *CNSE* is the central tool in a specific scenario called “a multi-site master course between KTH and UPC” proposed to show how Grid4All modules, services and applications will be deployed on the computers in the grid. This scenario represents a master course in which students should learn how a Distributed Hash Table works on a network environment. The proposed scenario uses the *CNSE* to run parameter-sweep simulations in a network laboratory in two educational institutions. Additionally, any institution belonging to the democratic grid could perform a similar educational scenario using the *CNSE*.

The first part of this course consists on the following steps:

- Educator provides course material including input scripts for *ns-2*, implementing several DHT scenarios.
- Students prepare network topologies and select parameters to run test simulations.
- Students perform parameter-sweep simulations and get the results to visualize them, opening a collaborative discussion.

The aforementioned steps can be supported by the CNSE and a Virtual Organization composed by some of the Grid4All members. In this VO, computing and storage resources must be shared regarding the needs of the Computing Network Course.

In the previous lines, a sample scenario has been presented. Unfortunately, the real scenario is a little bit complex because students have to work in presence of churn (it means that nodes where services have been deployed can fail, leave or join). Thus, CNSE can take advantage of the DCMS (Distributed Computer Management System) to achieve *self-* management* capabilities (self-healing, self-optimizing, and self-configuring). In fact, the least failure tolerant service is the *Simulation Service*, because once the *Broker Service* splits a parameter-sweep simulation in a set of *jobs*, all of them has to be completed before the merging of the results and the return back the *Broker Service*. In this context, DCMS must monitor the nodes in which *Simulation Services* has been deployed to detect failures and report the *Broker Service* to re-send this *job* to another available *Simulation Service*. Although the code for the integration of DCMS and CNSE has not been developed, some theoretical approaches has been discussed in the Grid4All to integrate the *Simulation Service* as part of the DCMS components.

Besides, CNSE can integrate VOFS (*Virtual Organization File System*) to support the storage requirements from the *Shared Repository Service*. The reason is that the simulation packages, including input scripts, XML manifest, output files, and maybe some additional files, can be large enough to require the integration with this *middleware*.

Moreover, CNSE can integrate VOMS (*Virtual Organization Membership Service*) to support the requirements for collaborative identification and monitoring from the *Group Context Service*. CNSE can also integrate SIS (*Semantic Information System*) for the registry and discovery of services. This service is especially relevant when the *Broker Service* has to discover the available *Simulation Services*, and *Visualization Services* to split the simulations into *jobs* and represent the output results. Thus, SIS could replace the *Index Service* facilitating the publishing of features that would be discovered semantically.

In general, the integration of the CNSE with these components has been only explored to the level of mapping of API calls, and discussion with developers. Nevertheless, it has not been implemented yet, because the priority has been the development of an application that could be used in a simplified real setting. As a result the required glue code needed to link CNSE with the Grid4All services has not been developed. However, that effort would simply be a matter of writing glue code with no expected significant obstacles.

To sum up, CNSE is a specific network simulation environment designed to be employed in educational scenarios for Computer Network courses. The current prototypes allow users to make parameter-sweep simulations including some collaborative features such as collaborative visualizations and collaborative storage. The next steps would be the development of the glue code for the integration of the *Simulation Service* with DCMS, the *Shared Repository Service* with VOFS, and the *Group Context Service* with VOMS. Additionally, the *Index Service* would be replaced by the SIS.

The future lines include the specific evaluation of the overall context in a real scenario in which tools are integrated with middleware, and students and educators play the main roles. With respect to this, a previous evaluation has been carried out for the CNSE with very satisfactory results.

Deliverable D4.3 and D4.4 describe the application architecture, operation and environmental conditions. D4.8 provides more details on the final prototype of CNSE and its user manual.

6 Shared Calendar (Sakura)

Sakura is a shared calendar application. It allows users to collaboratively organize and manage meetings. The collaboration is asynchronous. Sakura maintains the invariant that the user is not double-booked into two different meetings at the same time.

Sakura relies on Telex for replication, conflict detection and resolution, and eventual commitment.

We designed the Sakura application as a representative of collaborative applications. In fact, Sakura targets a human end-user and fits the Grid4All vision as it supports sharing replicated data following an optimistic execution model. Beside, Sakura integrates rich collaboration semantics, such as high level conflicts (double-booking), group and alternative meetings (explained shortly).

Thus Sakura design helped us capturing decision-making collaborative application requirements, and accordingly defining Telex API to support them. Sakura also serves as a model for future collaborative applications designed over Telex. It’s current usage is an evaluation prototype for Telex.

The final Sakura prototype provides the following features:

- Create a meeting
- Create a set of grouped meetings, meaning defining of a number of meetings scheduled on a separate dates but managed as a whole.
- Create a meeting with alternative dates.
- Invite a user to a meeting
- Change the date of a meeting.

Note that only Telex rich semantics enables Sakura to support grouped meetings and ensure eventual consistency even when a user is tentatively engaged in several meetings. Those are features that other calendar applications do not support.

Sakura application can be involved in the Grid4All educational scenario, where teachers and students in a VO share their calendars and use Sakura to organize conflict-free teaching sessions.

Current Sakura prototype assumes that collaborating users are authenticated, have correct writes on the shared calendars, and enough storage space. However, Sakura could integrate with VOFS (Virtual Organization File System) for storage services and VOMS (Virtual Organization Membership Service) to support authentication requirements.

The concurrent development and design of Telex limited the development of Sakura. Therefore the disconnected mode is not tested, and the current implementation stores all calendars in a global database. An alternative implementation maintains a separate calendar document per user, replicated only where some user needs it. Such partial replication is challenging, but is Telex responsibility, and transparent for the end-user.

The next step is to complete the prototype to support partial replication, and evaluating its impact on scalability.

7 Conclusions

This deliverable reports the final results on the development of applications in work package 4. The application prototypes can be found at the project web site[1].

These applications have been useful as ways to discuss and evaluate the intersection between the support required by end users in diverse collaborative tasks (scheduling of meetings, production of joint documents, simulation, media conferencing) on the educational scenario, the dynamic environmental conditions that the project is targeting (churn, dynamics, volatility, heterogeneity, scale), and the middleware components (Telex, VOFS, VOMS, DCMS, and indirectly the remaining components). However, the integration of Grid4All components into the applications prototypes has reached diverse levels depending on the limited resources available for each, the obstacles found, the effect of the concurrent development of the Grid4All services with the applications, the priorities for development, and the time available.

[1] http://www.grid4all.eu

8 References

D4.1 Specification of scenarios, user requirements, and infrastructure requirements; Grid4All project deliverable (M12), 2007.

D4.2 Specification of situations derived from applications; Grid4All project deliverable (M12), 2007.

D5.1 Evaluation and test plan; Grid4All project deliverable (M12), 2007.

D4.6 Integrated scenario and architecture; Grid4All project deliverable (M18), 2008.

D4.7 Measurable requirements including end-user requirements; Grid4All project deliverable (M18), 2008.

D4.3 Prototype API usage by Grid4All based applications; Grid4All project deliverable (M24), 2008.

D4.4 Initial prototypes of all applications; Grid4All project deliverable (M24), 2008.

D5.4 Evaluation criteria and test plan; Grid4All project deliverable (M24), 2008.

D5.3 Final integrated proof-of-concept implementation and evaluation report; Grid4All project deliverable (M36), 2008.

D4.8 User Manual; Grid4All project deliverable (M36), 2008.

Level of confidentiality and dissemination

By default, each document created within Grid4All is © Grid4All Consortium Members and should be considered confidential. Corresponding legal mentions are included in the document templates and should not be removed, unless a more restricted copyright applies (e.g. at subproject level, organisation level etc.).

In the Grid4All Description of Work (DoW), and in the future yearly updates of the 18-months implementation plan, all deliverables listed in section 7.7 have a specific dissemination level. This dissemination level shall be mentioned in the document (a specific section for this is included in the template, both on the cover page and in the footer of each page).

The dissemination level can be defined for each document using one of the following codes:

PU = Public

PP = Restricted to other programme participants (including the EC services);

RE = Restricted to a group specified by the Consortium (including the EC services);

CO = Confidential, only for members of the Consortium (including the EC services).

INT = Internal, only for members of the Consortium (excluding the EC services).
This level typically applies to internal working documents, meeting minutes etc., and cannot be used for contractual project deliverables.

It is possible to create later a public version of (part of) a restricted document, under the condition that the owners of the restricted document agree collectively in writing to release this public version. In this case, a new document code should be given so as to distinguish between the different versions.