Project no.: 034567

Project acronym: **Grid4All**

Project title: **Dynamic Virtual Organizations for schools, families and all**

Instrument: Specific Targeted Research Project (STREP)

Thematic Priority: Information society technologies (IST, Thematic Priority #2)

D7.5a – Publishable Final Activity Report

Period Covered: from 01/06/08 to 30/06/09

Date of preparation: 4/08/2009

Start date of project: 01/06/06

Duration: 37 months

Project coordinator name: Ruby Krishnaswamy

Revision: R1

Project coordinator organisation name: FT

Contractors involved

This document is part of a research project partially funded by the IST Programme of the Commission of the European Communities as project number IST-FP6-034567. The partners in this project are:

- France Telecom (FT, coordinator), France
    - Contact person: Ruby Krishnaswamy ruby.krishnaswamy@orange-ftgroup.com

- Institut National de Recherche en Informatique en Automatique (INRIA), France
    - Contact person: Patrick Valduriez patrick.valduriez@inria.fr

- The Royal Institute of technology (KTH), Sweden
    - Contact person: Vladimir Vlassov vladv@kth.se

- Swedish Institute of Computer Science (SICS), Sweden
    - Contact person: Per Brand perbrand@sics.se

- Institute of Communication and Computer Systems (ICCS), Greece
    - Contact person: Nectarios Koziris nkoziris@cslab.ece.ntua.gr

- University of Piraeus Research Center (UPRC), Greece
    - Contact person: Symos Retailis retal@unipi.gr

- Universitat Politècnica de Catalunya (UPC), Spain
    - Contact person: Leandro Navarro leandro@ac.upc.edu

- ANTARES Produccion & Distribution S.L. (ANTARES), Spain
    - Contact person: Sergio Lopez Borgonoz sergio@antares.es

Contact

Ruby Krishnaswamy, Grid4All coordinator
France Telecom, France
Email: ruby.krishnaswamy@francetelecom.com
Website: www.grid4all.eu

Table of Contents

1. Project Execution

1.1 Summary description of project objectives

The Grid4All project was formulated almost 4 years ago. At that time, grids represented the state of the art in general-purpose distributed systems, and had some of the features required for democratic scenarios, such as sharing and aggregation of resources, and forming Virtual Organisations (VOs) across organisational boundaries. Our vision was naturally expressed as a generalisation of grids, a movement from Grid-For-Some to Grid4All. Today, an appropriate title might be Hybrid, Democratic Cloud.

The project was set up to enable open, democratic access to computing resources, supporting both grid and edge resources, and to support co-operation within distributed groups of users, so-called “virtual organisations.” The name Grid4All, and its slogan “democratic Grid,” highlight its focus on supporting small users, having limited financial, physical, and system administration resources. To these users, important factors are openness, the ability to utilise all available resources (including volatile ones), ease of use and selfmanagement. The original project objectives may be divided in two themes (a) pooling resources and utilising them effectively and (b) collaboration and sharing data.

Pooling resources and utilising them effectively:

Autonomic configuration and management for dynamic virtual organisations: A key requirement for Grids targeting domestic users on the large scale on the Internet is to alleviate the current need for massive administration of the Grid. We not only want to develop a Grid that largely is self-organising and selfrepairing, but we clearly need to. We are targeting ordinary users, families, and small organizations where we cannot expect to find the time, money, or expertise for extensive management functions.

Utility model: The Internet and its users constitute an infinite pool of resources and services. One aspect of democratization is to enable resource owners to share their resources in fair ways. Rigid and centralised businesses such as Amazon do already exist but if any Internet user were able to propose his/her resources in a secure and advantageous way, a new type of open market will emerge with economic advantages: better competition, more accurate supply/demand adequacy. This also has sustainable development aspects since harnessing of captive idle resources implies reduction of wastage.

Resource and service discovery: SOA and Web Services offer a standards-based interoperability platform and allow organisations of all kinds to integrate their applications and improve accessibility to end users. To allow software and humans to find available services and ensure that services interoperate, services require visibility. Registries are about this, to publish and discover services. Services may be advertised and queried using different vocabularies even though they may mean and act the same. More so in the case of democratic grids where a large number of independent actors may publish and discover services. Registries based on simple syntactic matching offer limited understanding between providers and consumers (peers), each peer having its own language. Semantic approaches to discovery where matching is based on formal descriptions of services overcome this restriction by providing a common understanding. Furthermore, ranking of discovered services according to peers' preferences, rather than only to their "semantic distance" from their requirements contributes to shifting towards personalized discovery services.

Collaboration and sharing data:

Different collaboration styles: Information sharing is a basic requirement of any VO. Existing grid-oriented file systems are not designed for collaboration, do not scale well, and do not ensure application-specific levels of consistency. Existing collaborative editing tools do not help with high-level conflict detection and reconciliation.

We want to be able to support (in the form of tools, models, and basic services) the loosest, most relaxed consistency models that suffice for the needs of any given application. This is crucial in that we target communities that are severely more limited in terms of bandwidth, connectivity, and reliability on the level of the individual machines than in standard Grids.

The core objectives of the project:

To summarize, the goals of Grid4All is to enable open, democratic access to computing resources, supporting both grid and edge resources, and support co-operation within distributed groups of users. As a research project, Grid4All focused on advanced technological enablers: peer-to-peer based autonomic management facilities, resource market, and federative data storage and access. The core objectives to attain project goals were the following:

- Middleware and tools for autonomic configuration and management: actions such as the deployment of a Grid application, the reconfiguration of Grid services and application in case of a node failure or the resizing of the Grid pool of allocated resources in case of workload increase should be handled transparently by the infrastructure. An autonomic software framework based on a component model provides a high level of abstraction for deploying, configuring and reconfiguring at runtime Grid applications or services.
- Tools for new collaboration styles: Frameworks supporting large-scale sharing of data within virtual organisations should allow a user to easily create and administer VO-specific workspaces. Users should be able to expose information and storage resources to one or several workspaces. They should be able to provide names for exposed objects that can be communicated across the VO, to access data and to maintain different degrees of consistency according to the object's specification.
- Advanced service matchmaking: Grid4All will investigate more user-specific query formulations and service selection. This is in contrast to traditional Grid systems where typically the discovery services match user specified attributes and properties. The objective is the development of a centralized service for the discovery of markets that trade requested/offered resources configurations, as well as markets that trade services provided by service providers, and services provided outside from market places. Scalability and availability is also a concern and designs for decentralized architectures will be studied.
- Infrastructure for trading resources: To allow Grid4All virtual organisations to allocate computational resources in a cost-effective and flexible way, the objective is to design a resource market place that enables trading of computational resources on the Internet. The market place software infrastructure should be open and extensible. Given the global scale expected of the marketplace a centralized allocation scheme will prove infeasible and thus some form of decentralized allocation must be used.
- Groupware applications based on the Grid4All platform: End users perceive the usefulness of the platform indirectly by the functional and non-functional behaviour of applications and the environment within which the applications are capable of executing. It is clear that the applications that are adopted by the project do already exist, but these currently execute within controlled environments - expensive infrastructures, high level of administration, expensive software. To demonstrate the feasibility of the Grid4All software platform in addressing requirements of a democratic Grid we will design and develop groupware applications using Grid4All technology enablers through different application prototypes.
- Evaluation and demonstrators: The implementation of the architecture should be put into practice in real-world case studies. Grid4All should illustrate in practise its impact on new methods for real-world applications and scenarios. While evaluation of complete integrated architecture is beyond the scope of the project, the objective is to implement meaningful integration scenarios, which show advanced capabilities by combining different functional components. Qualitative evaluation to obtain indicators on the level of acceptance must be conducted.

1.2 Methodologies and approaches

To achieve its objectives, the project developed prototype implementations and the execution of the project was guided by the following principles:

- **Architecture-based management approach**:

  Grid4All follows the architecture-based management approach, which uses architectural models as the basis of deployment, monitoring and reconfiguration activities. This approach provides a high abstraction level for management, hiding the idiosyncrasies of the underlying infrastructure. It enables seamless integration between configuration management and deployment activities, limiting architectural erosion and enabling a more rapid development/deployment cycle. Design and development of applications and services relies on the extended Fractal component model (implemented using Niche) enabling "write-once deploy-anywhere" self-managing distributed applications.

- **Integration based on innovative scenarios**:

  Grid4All has designed and developed a set of software modules that commonly address the overall project objectives. While complete integration of all software modules is clearly infeasible, design and development have followed common architectural principles. Integration of different components provides enhanced capabilities; e.g. VOFS integrated to the Grid4All security architecture enforces access control on shared workspaces. The project has identified and implemented the most innovative integrative scenarios.

- **Prototyping of new applications and adaptation of existing applications**:

  The project aims to enable "democratic grids", i.e. to provide new classes of users with flexible access to large computational capacities, to enable them to pool their computational resources over the network, and to empower them with sharing and collaboration facilities. Middleware, services and tools are designed to work around the problems intrinsic to such grids (volatility, churn, heterogeneity etc) and to users of such grids (different levels of IT awareness and skills, disconnected mode of work etc). Applications are not necessarily new in terms of functionality, but need to execute and provide services even when deployed in such environments and for and by such users. To evaluate Grid4All technology results, we need to not only prototype new applications but also adapt (as minimally as possible) existing applications.

- **Evaluation**:

  Qualitative evaluation procedures that the project had defined allowed obtaining the perspective from end-users and their appreciation of the functional and non-functional aspects of the system. We had identified two categories of evaluators: end-users who install, set-up and use the software product and developers who use the product (middleware) to design and develop new applications. The first category applies to VOFS, CFS and eMeeting and the second category to Telex and Niche. Evaluators with different levels of experience and knowledge were selected. This helped to obtain 'progressive' feedback; skilled evaluators were able to provide advanced feedback related to innovative functionality whereas less experienced evaluators gave their appreciation on the ease of use, quality of documentation etc. Quantitative evaluation procedures sought to measure metrics specific to individual functional components. The evaluation work consisted of design of test cases, planning and running experiments, collecting and analyzing results. Testing was conducted either on distributed platforms such as PlanetLab and Grid5000 or on single compute nodes (according to functionality evaluated). Intermediary evaluations helped improve design and implementation during the course of the project, while final evaluations will guide future work.

- **Case studies**:

  The implementation of the architecture will be put into practice in real-world case study, serving both as a proof of concept for the architecture in all of its scenarios, and also as a vehicle to analyze empirical usage findings in a real-world context. While implementation of such a case study is beyond the scope of the project, two consortium partners (UPC and KTH) have proposed an Erasmus Mundus project to conduct a multi-site course on distributed systems. This proposal submitted in April 2009 (and approved) describes a real-world scenario and is a natural domain of application for Grid4All results.

1.3 Activities performed and results

1.3.1 Management Tools

Managing distributed applications operating on large-scale, dynamic environments, such as democratic grids or clouds, is complex and error-prone. Application management generally consists of two parts. First, there is the initial deployment and configuration, where individual components are shipped, deployed, and initialized at suitable nodes, then the components are bound to each other as dictated by the application architecture, and the application can start working. Second, there is dynamic reconfiguration where a running application needs to be reconfigured. This is usually due to environmental changes, such as change of load, the state of other applications sharing the same infrastructure, node failure, node leave (either owner rescinding the sharing of his resource, or controlled shutdown), but might also be due to software errors or policy changes. All the tasks in initial configuration occur also in dynamic reconfiguration. For instance, increasing the number of nodes in a given tier will involve discovering suitable resources, deploying and initializing components on those resources and binding them appropriately. However, dynamic reconfiguration generally involves more, because firstly, the application is running and disruption must be kept to a minimum, and secondly, management must be able to manipulate running components and existing bindings.

To reduce the complexity of those tasks, Grid4All proposes a set of management tools based on the *component-based management* approach; the tools include the *Niche* platform, an ADL-based *deployment service*, and the *DepOz* framework.

Component-based management

In the component-based management approach, all system elements are constructed or wrapped as components and management takes the form of inspecting and modifying the system architecture containing components, their interconnections and their attributes. This approach provides a high abstraction level for management, hiding idiosyncrasies of heterogeneous application elements and management techniques. For example, complex multi-tiered applications comprising diverse legacy software (e.g. web servers and databases) may be managed in a unified way. It integrates seamlessly the different phases of application lifecycle; design, deployment and operation, each modifying and producing component configurations.

The choice of component model is crucial in architectural approaches to management. Grid4All adopts the Fractal component model, a language-agnostic model with support for building reconfigurable systems. A key Fractal feature is the hierarchically assembling of components into composite components. Hierarchical composition reduces complexity of designing, developing and reconfiguring systems. For example, a multitiered application can be modelled and manipulated as a single component containing components representing each tier.

Niche

Niche is a general-purpose distributed component management system (DCMS) used to develop, deploy and execute self-managing distributed applications or services in all kinds of environments, including very dynamic ones with volatile resources. Niche is both a component-based programming model that includes management aspects as well as a VO-wide distributed run-time environment. Niche builds on work on autonomic computing and is Java- and Fractal-based, where Fractal defines the management interfaces of components. The control loop paradigm is used where management logic in a continuous feedback loop senses changes in the environment and component status, reasons about those changes, and then, when needed, actuates, i.e. manipulates components and their bindings. A self-managing application can be divided into a functional part and a management part tied together by sensing and actuation.

Non-standard and novel features in Niche: Niche is a VO-wide infrastructure that loosely binds all physical resources, and provides for resource discovery by using a structured overlay. Niche provides a sensing service for application managers, as part of its programming API. Application managers can subscribe to application-specific component events (e.g. load) or platform events (e.g. failure). Niche provides an actuation service, whereby managers can control and manipulate with components and their bindings. The sensing and actuation services are robust and churn-tolerant, and Niche itself is self-managing.

Niche allows for maximum decentralization of management. Management can be divided (i.e. parallelized) by aspects (e.g. self-healing, self-tuning), spatially, and hierarchically. A single application, in general, has many loosely synchronized managers. Niche also provides the execution platform for these managers; they typically are assigned to different machines in the VO. There is some support for optimizing this placement of managers, and some support for replication of managers for fault-tolerance.

An important feature is that all elements of the architecture, components, managers, etc., have VO wide identifiers. Niche ensures that actuation messages reach the component and sensing events reach (at least once) the subscribed manager, even if the component or manager has moved.

ADL-based deployment service and DepOz

Programming in Niche is based on a low-level, Java API. It is therefore desirable to program management at a higher level. To this end, the management tools include a deployment service driven by declarative application descriptions written in ADL (Architecture Description Language). The descriptions contain the logical application architecture in terms of components and interconnections as well as the resource requirements of individual components. The descriptions contain no infrastructure details, thus allowing deploying the same application on different infrastructures. The deployment service translates these descriptions to Niche operations for setting up the initial application configuration.

The ADL-based deployment service is sufficient for initial application deployment but not for dynamic reconfiguration. To provide high-level and user-friendly language support for dynamic reconfiguration, we have developed the DepOz framework. DepOz is based on the Fractal component model and the Mozart/Oz distributed environment and supports two main capabilities. First, it supports navigating and querying component structures, much like XPath is used to navigate and query XML documents. However, unlike XPath, DepOz expressions capture the dynamicity of component structures and their values are automatically updated following architecture evolutions. Second, DepOz supports defining reconfiguration workflows in a concise and compositional way. DepOz directly supports all common workflow patterns (e.g., parallel split and synchronisation), and it can be easily extended to capture new patterns.

DepOz is not currently integrated with the Niche platform. It was implemented in the Oz language as a proof-of-concept research prototype in order to provide and validate high-level language and tool support for self-configuration. Integrating DepOz with Niche will further simplify the development of distributed self-managing applications for volatile environments.

Rationale of Approach: In the Grid4All vision, non-professionals should be able to form a VO to collaborate and share their resources. In that VO, they should be able to install, deploy, use and manage nontrivial distributed applications with minimal effort. This requires that the applications are self-managing. Furthermore, in these environments, resources are at their most volatile, and therefore standard mechanisms to achieve self-management will not work. A single management node that continuously monitors the entire system and keeps an up-to-date system map introduces not only a single-point-of-failure but also more importantly a bottleneck. The more dynamic and volatile the system is the more environmental events will be generated. Not only does Niche support decentralized management, but the sensing/actuation infrastructure has features to reduce messaging. For instance, a manager that is responsible for self-healing does not need to be informed about component movement due to, for instance, a node shutdown. Indeed, in a dynamic Grid4All VO, system components might be reshuffled in the continuously changing resource pool for some time before a failure occurs and the self-healing manager is triggered. In addition, the Fractal model was extended to include component groups with one-to-all, one-to-any bindings for maximum efficiency.

Relevance to Clouds: We believe that the management tools and the principals of how they were built are extremely relevant for hybrid clouds, as well as democratic grids, as most of the issues are similar. In particular, it makes economic sense for users to integrate their own infrastructure with a cloud, generally running their own infrastructure at full capacity and using the absolute minimum of cloud resources for peak and excess demand. This introduces the kind of volatility with cloud resources being grabbed upon need and then being released as soon as possible, that Niche was designed to handle.

Fortuitously, our work might also be relevant for data-centre clouds. In developing Niche, our guiding principles were churn-tolerance, decentralization, and, in principle, scalability. (Though as we were not targeting large scale systems there are non-scalable aspects in the Niche implementation, but these are engineering issues). According to Birman et. al.[1] an important lesson learned by the cloud computing community is the importance of churn-tolerance and decentralization (scalability too, but that is given). Unfortunately, as was also pointed out in the article, it is virtually impossible for the research community to test mechanisms in large-scale systems (hopefully, this will be addressed in the European FIRE initiative[2]). Niche and our two application demonstrators (YASS and YACS[3]) do illustrate churntolerance and decentralization of management in small-scale systems. We do not know what would happen on the scale of clouds. For instance, Birman points out that in principle self-stabilizing mechanisms can exhibit unacceptable oscillations when scaled up.

1.3.2 Collaboration

Sharing files is a basic task of a co-operating group of users. According to our "democratic" vision, setting this up should be a casual operation and should not require any central authority or infrastructure. To that end, we designed VOFS, a virtual-organisation oriented file system.

VOFS provides each user with their own file system to share in the network. Users are able to federate views of their respective file systems, and to create workspaces of shared storage space and file content. Federating is straightforward and remains decentralised in the style of the World-Wide Web: users simply set up links to identify shared resources.

For applications, VOFS is transparent, in the sense that files are accessed via the standard POSIX interfaces. Additionally, applications may exploit the additional VOFS features, such as federation, caching, disconnected operation or notification services and generic messaging.

Users remain in strong control over their resources. The file systems of different users are independent; federation of files in shared workspaces does not compromise this. The access control architecture is flexible, as it allows users to program their own access policies. For instance, a policy might refer to a central VO security infrastructure.

---

[1] Ken Birman et al. Towards a cloud computing research agenda. In ACM Sigact News, pages 68-80. ACM, June 2009. Report on Workshop on Large Scale Distributed Systems (LADIS 2008).

[2] Objective 1.6 ICT Work Programme. Future internet research & experimentation, June 2009. http://cordis.europa.eu/FP7/ict/fire/

[3] http://niche.sics.se

VOFS is designed to maintain functional independence between users, despite failures of the network or of remote computers. Caching is used to hide network latency, to provide disconnected access, and more generally, to avoid disruption from network events. This makes the file system very responsive, and is especially valuable in loosely coupled collaborative environments. By design, concurrent access to the same file are not synchronised, and a best-effort approach is taken. When stronger guarantees are required, applications should use the Telex Semantic Store.

While users are independent from authorities and are able to create their own workspaces around their workflows, third parties may also enter the network to offer their services, by launching a VOFS peer to represent them. The peer appears as another regular peer to the users. The service provider may extend VOFS to install their own logic behind serving files or storage without creating any disruption or dependency to other users. For example, a library might export a corpus of articles to a group of collaborating researchers, in the form of a VOFS file system. Furthermore, the library may extend the file system with "virtual paths" that instantiate library queries. For instance, virtual directory */search/author/Turing* would contain all documents whose author is Turing. Applications access it just like any other directory. No single peer is critical: if the peer hosting the library crashes, the only disruption is the temporary loss of the library service; other services or files are not affected.

The resulting environment combines casual and independent user collaboration with potential access to big, organised third party services in the cloud. Cloud products such as *dropbox.com* or *box.net* provide very simple ways of sharing information, *wuala.com* leverages social networking, but users cannot create their workflows beyond what is given by the service provider. In contrast, the casual, unco-ordinated character of the World-Wide Web inspires our approach where large corporate providers co-exist with individual users, and the latter retain their freedom and control. Like the Web, VOFS democratically empowers users; but where the Web only supports posting content to unlimited readers, VOFS supports true co-operation inside defined groups.

Co-operative group work in Grid4All

To support effective co-operative group work requires well-defined consistency guarantees, beyond the best-effort approach of VOFS. The Telex Semantic Store addresses this issue. Telex also addresses the larger problem of providing both scalability and well-defined consistency guarantees over shared mutable data. This will enable novel applications that require consistency guarantees to operate over cloud-scale infrastructures.

Remote and offline data sharing are increasingly important. Examples include a shared wiki, cooperative offline editing with Google Gears, enterprise platforms such as Notes or Groove, collaborative code repositories CVS or SVN, and so on.

To deal with failures, latency, and large scale, a common approach is optimistic replication (OR). OR decouples data access from network access: it allows a processor to access a local replica without synchronising. A site makes progress, executing uncommitted actions, even while others are slow or unavailable. Local execution is tentative and actions may roll back later. An OR system propagates updates lazily, and ensures consistency by a global a posteriori agreement on a same state.

Implementing a collaborative application is complex. The programmer needs to manage asynchronous communication between collaborators, data replication, conflict detection and resolution, and eventual agreement. Besides, end-users should be able to make sense of the collaborative work: (i) they should not be overwhelmed or confused by remote updates and rollback, (ii) conflict detection should be relevant, (iii) and users should be able to express preferences for conflict resolution. Additionally, the application needs to be responsive, and to provide guaranties about the data persistence and convergence.

To solve these issues, current collaborative applications use ad-hoc approaches that do not guarantee correctness, and leave manual work to the user. In contrast, we propose an open platform called Telex that helps developers to build collaborative applications faster. Telex supports an optimistic replication model for sharing stateful data in a decentralised way over a large-scale network. Telex eases application development by taking care of the communication, replication and consistency issues. Based on a principled approach, Telex guaranties that replicas never violate safety and converge eventually. The Telex platform is designed for robustness, flexibility and performance.

The Telex middleware layer ensures that different network nodes may independently update their local replica of shared data, without prior synchronisation. This supports disconnected operation and cooperative work, and helps general scalability. Each user may have a different tentative view of the state of shared data; Telex provides well-defined guarantees that programs may rely upon: replicas eventually converge, and application invariants are never violated, despite concurrency, disconnection and reconciliation. Example applications include a consistent, peer-to-peer, shared calendar system, Sakura and a co-operative Wiki editing system.

To scale up, clouds must avoid strong consistency mechanisms and promote decoupling [Ladis]. Indeed, the Telex consistency mechanisms are parameterised with the semantics of shared data items, and ensure scalability by relaxing consistency mechanisms and moving them to the background. The main features of Telex are:

Collaborative application development faster and easier: Telex provides a facility of sharing by taking care of complex application-independent aspects, such as replication, conflict repair, and ensuring eventual commitment. This clear separation of concerns allows the programmer to concentrate on core functionality.

Work in disconnected mode: Telex handles disconnection transparently for the application. When a user is offline, Telex continues logging his updates locally, and the user can keep working on the shared document. When he reconnects, Telex automatically sends the local updates, fetches the missing ones, and checks for possible conflicts.

Multidocument / multi application sharing: Although documents are the basic sharing unit, Telex does not assume that documents are always independent. Instead, it allows a user or an application to define a constraint between actions of two distinct documents, in order to enforce consistency across documents. This is possible even if these documents are processed by distinct applications as applications share the same engine. Consequently, novel cross-application scenarios are possible.

Ergonomic and more productive collaboration: Telex implements several mechanisms that enhance collaboration ergonomics and productiveness:

- **Filtres**: Telex allows users to set filters defining their own view of a shared document. For instance, they should see each other updates in real time in the case of shared whiteboard. On the other hand, they can select a personalized view where other updates are ignored when editing a text document, in order not to be overwhelmed by group activity.
- **Smart conflict detection and resolution**: In order to be relevant, conflict detection and resolution consider fine-grained semantics. All possible levels of semantics can be addressed: data semantics, application semantics and user intents. Besides, Telex conflict resolution algorithms implement heuristics to decouple independent conflicts and computes optimal solution, minimizing the amount of lost work. Thus, Telex provides the application with alternative solutions grouped by conflict, and sorted by quality.
- **Social collaboration**: Telex enables users to express preferences for conflict resolution and agreement. For instance, they should be able to retain the most important update, or minimize the amount of lost work.
- **Guarantees**: Telex protocols are proven correct. They ensure that replicas never violate safety (concerning the provided constraints) and converge eventually.

1.3.3 Connecting providers and consumers

In a democratic Grid, finding the right resource is a challenge. This is complex, since such a large-scale, unstructured system offers a large number of providers of various sizes and specialisations. They will offer a vast, dynamic array of services and resources under largely differing conditions and SLAs. Furthermore, any user's demand may dynamically expand or shrink, depending, for instance, on the demands of specific applications. This requires an effective matchmaking service between requests and offers for services and resources. Indeed, such a service is already a vital requirement in grids.

The mutual discovery of clients and providers is an essential feature of the open market of the hybrid cloud, where resources are traded as goods and subject to supply and demand. Resources are made available through spontaneous agent-initiated market services. Resource matchmaking generalises to the problem of discovering those markets, each with its own auction mechanism.

The Grid4All experience suggests a declarative, semantic approach. Semantic descriptions have the potential to support automation of service retrieval, invocation, composition and monitoring tasks by providing machine-exploitable, meaningful declarative descriptions of service characteristics.

Reducing complexity in negotiation-based resource allocation

Many large-scale platforms have adopted economic and market-based means to allocate resources, typically adopting bilateral interaction between interested parties, i.e. providers and consumers. Such platforms do not scale with numbers of participants, applications, resources and are complex to manage; participants need to conduct multiple sequential or simultaneous negotiations to obtain all resources. We propose a mediating platform, i.e. auction-based market place to reduce complexity.

G4A-SIS is a major component of the Grid4All open market-place. Ontology has been designed to support atomic unitary ICT resources, and aggregations and composites. Resources are traded at formally conceptualized auction-based markets. Offers and requests describing simple, aggregate or composite resources are traded at auction-markets and published at the G4A-SIS. Applications requiring resources query the G4A-SIS to locate auctions where matching resources are traded. Matches are done based on characteristics of the ICT resources and properties of the markets where such resources are traded.

Simplifying the usage of rich semantic technologies

Semantic information systems improve quality of search results, but to enhance acceptance, they should (a) hide low-level details of semantic technologies from users and (b) provide means to reuse the large quantity of legacy WSDL descriptions.

G4A-SIS simplifies the job of human annotators, i.e. service developers by providing automatic methods to exploit domain ontologies, service specifications written in WSDL, descriptive information such as API documentation, license agreements and other information sources. Textual descriptions are exploited to annotate, classify and assess similarities between Web Services. In cases where lexical items in WSDL specifications are scarce, state-of-art algorithms are used to synthesize annotations.

Humans may simply provide textual annotations in the WSDL specifications through natural-language descriptions. The semantic annotation using Ontology elements is done automatically and the employed algorithms provide a high quality of annotation. The developer retains control and may validate the generated semantic specifications.

Automatic annotation also reduces errors and improves usability, i.e. improves quality of matching and discovery. Furthermore, the automatic annotation designed in G4A-SIS avoids pitfalls related to phenomena such as synonym and homonym terms, typographic variations of terms, differences between granularity of services and ontological specifications. It also takes into account scarce and often misleading comments, descriptions and tricky names of involved service parameters being involved, as well as improper or faulty use of domain terminology.

Meaningful search results

When querying for services based only on functional constraints, a large number of similar services that fulfil the consumer's requirements may be retrieved. It is up to the consumer to wade through the replies before binding to a specific service. Semantic matchmaking is the first step, semantic similarity functions are employed to identify services matching queries. The G4A-SIS also allows selection of only those services that provide the desired functionality and that satisfy desired criteria and preferences of users.

Matched queries are ranked using both consumer and provider preferences. The Satisfaction-based Query Load Balancing (SQLB) framework of the selection component of G4A-SIS preserves both consumer and provider preferences and interests. In large-scale information systems, query allocation, i.e. the method of allocating consumer requests to providers that may satisfy the request, often privileges overall system performance, e.g. balances load across providers. The method that is used here incorporates consumer interests as well.

Summing up the above, aiming to operate at a large scale with multiple providers that have varying resources and services provision abilities, accentuating the role of the supply and demand law, hybrid clouds need effective matchmaking services between offers and requests for services/resources.

The Grid4All Semantic Information System provides a first step towards effective matchmaking in a large-scale system subject to dynamic offers and dynamic demand. For reasons of expedience, the current implementation is centralised. One of the major lessons learned is that, in order to remain effective in such a large-scale, dynamic environment, the registry must be decentralised. A distributed version of G4A-SIS has been designed to achieve this goal.

1.3.4 Utility model for democratic grids

Internet computing, volunteer computing, global and desktop computing are different terms used to designate computing on collections of heterogeneous computers scattered around the globe. In Grid computing, participating nodes are often clusters (of compute nodes), whereas in global computing, nodes are mostly individual and rarely small sets of networked computers. It bears the name 'volunteer computing' since individual owners voluntarily donate their idling capacity to projects such as folding@home.

Peer-to-peer systems came up to enable content sharing. Free riding; a frequent phenomenon is dissuaded through accounting mechanisms that ceil download requests as a function of upload capacity offered by the peer. Characteristics of such systems are (a) volatility and lack of stability, i.e. nodes may disconnect at will (b) heterogeneity (of nodes, communication bandwidth and latency, networks) and (c) large scale.

These systems do sufficiently encourage participation in our idea of global and democratic computing where just anybody on the network may seek ICT resources. Contrary to traditional Grids and grid virtual organisations, users of democratic grids are self-interested. There is a need for incentive mechanisms in democratic grids and the best candidate seems to be economically rewarding mechanisms.

Market places allow buyers and sellers to carry out trade using well-defined rules whose ultimate effect is to adjust prices. By adjusting prices, supply and demand are balanced. We provide tools to support building resource market places, focusing on two aspects [8]: (i) the allocation and pricing problem (ii) scalable dissemination of price signals to shape participants decisions. Together they provide a way for decentralised coordination and self-organising resource allocation in democratic grids. Combinatorial auctions offer a competitive solution to the complex resource co-allocation and planning problem; but they are computationally complex and hard to interpret.

By supporting spatial and temporal partitioning of auctions, the market place will scale with number of participants. Interpretation of market state can be improved through suitable pricing schemes. Partitioning and decentralisation should not cause loss of information leading to inefficient allocation and price instability. Accurate price signals aid buyers and sellers to plan capacity, e.g., lease resource at times that maximise their gain or reorganise internal workload and sell capacity when demand is high. We address these through commodity pricing schemes and dissemination of approximate aggregated information.

A vocabulary and structure to describe ICT resources

Resources may be combined in arbitrary configurations to represent workflows. This structure, referred to as bidding language, is available in a machine-readable and extensible XML-based format. Non-functional and quality parameters may be expressed and the bidding language can be extended to application-specific services.

Two auction mechanisms, to trade resources, one for each of the two application classes cited above. The combinatorial auction allows expressions of the type "*I would like 6 units of VM1 between 10:00 and 15:00 for 2 hours each AND 100 giga of storage during the period of VM2 allocation. I am willing to pay 5 euros for this. If this is not possible, then allocate 3 units of VM3 between 11:00 and 13:00 for 1 hour each AND 4 units of VM3 between 13:00 and 15:00 for 1 hour each AND 160 Gb of storage during the period of VM2 allocation and 100 Gb of storage during the period of VM3 allocation. In the latter case I am willing to pay 6 euros*". An extended version of the well-known double auction with the k-pricing policy is suited for applications of the first category. It accepts expressions such as "*I would like at most 6 units of VM4 between 10:00 and 17:00 for at most 3 hours each. I am willing to pay at most 1 Euro for each unit of VM4 for each hour allocated.*"

Pricing Scheme

A pricing schema defines what the consumers pay and what the supplier is paid. It determines how the surplus generated by the market is redistributed. Pricing addresses multiple goals. (i) be incentive compatible (ii) allow participants to derive information on the real worth of the different resources over time in order to aid them in planning their capacity needs (iii) achieve market clearing, i.e. buying prices of losing bids are inferior to the computed price and reservation price of losing offers are superior to the computed price.

Consumers (or their applications) buy resources, either to execute the complete application workflow, or to provision extra resources on-demand for deployed applications in order to improve the quality of service. For example, a three-tier application may buy additional virtual machines for its logical layer in order to improve its throughput when request rate increases. Consumers need to decide when to buy resources, the optimal duration of resources, quantities and qualities. The pricing schemes that we propose generate per-resource prices as a function of time. The challenge in this is that buyers and sellers issue combinatorial requests; "*I need 3 Fast CPU and 120 Giga of Storage between 10:00 and 18:00 for a duration of 3 hours each, and am willing to pay of 5*" and similarly for providers. Extricating individual prices (of Fast Cpu or Storage) at specific times is a hard problem to solve, but even a reasonably good approximation eases interpretation for users.

Market Information Feedback

Resource providers strive to maximize earnings and consumers to maximize surplus, i.e. difference between the values that the service (or application) brings to the consumer and cost of leasing resource to execute them. Consumers may adjust quantity, quality, times and duration of resources by learning market prices of different resources at different times.

The Market Information System (MIS) developed in Grid4All collects information such as prices computed by auctions and routes aggregated information reliably and in a timely way to subscribers. Learning the market situation participants can refine their preferences regarding desired resource configurations and adjust them by analysis of previous and historical trade cases. MIS aids them in this decision-making by providing succinct information of prices: by category of resource and times. It provides market statistics such as volumes of supply and demand at different times of day, week, month etc. Consumers can plan capacity needs, by seeking resources at periods of low demand (low price) if convenient.

The challenge that we faced in the design of the MIS is that the market place is not centralized. Auctions may be instantiated on multiple compute nodes and at different times. Each instance operates for an interval of time and collects offers and bids from different providers and consumers. The auction clears and generates the statistics (prices, supply etc). The MIS should reliably gather all such information and report aggregated values to subscribers to aid in their decision-making. Our innovation is to provide approximate information and automatically balance time and accuracy. Users define a Confidence interval and the system adjusts the number of sources that is sampled for the aggregation.

As a conclusion, in the perspective of Cloud computing that allows consumers to adapt to surges by leasing more resources when demand spikes occur is now an oligopoly held by a few companies, which practise opaque pricing schemes. In a democratic cloud, the supply side should be democratised too. Any resource owner should also be able to sell its excess capacity. This encourages competition and innovation. Thus, the Grid4All Resource Market is an enabler for new business models, where entities that act as brokers of computing resources, will emerge. They will provide value-added services, SLA enforcement and risk management, capacity planning tools and reputation-aware market mechanisms.

1.3.5 Applications and scenarios

End users perceive the usefulness of the platform indirectly by the functional and non-functional behaviour of applications and the environment within which the applications are capable of executing. It is clear that most applications that are adopted by the project do already exist, but these currently execute within controlled environments - expensive infrastructures, high level of administration, expensive software. To demonstrate the feasibility of the Grid4All software platform in addressing requirements of a democratic Grid we have designed and developed groupware applications using Grid4All technology enablers in different application prototypes.

Usage scenarios derive requirements for infrastructure and applications for Grid4All intended uses. While some applications are designed from scratch, others are legacy applications that have been redesigned and adapted. This has helped understand the complexity imposed by dynamic environments on applications and define algorithms to overcome situations and limitations.

Adhering to our objectives, we have selected a range of applications of interest to end-user focus groups targeted by Grid4All. Collaborative File Sharing, eMeeting and Shared Calendar applications support collaborations; collaborative work is essential for teaching and research and for non-profit organisations, leisure and social networks.

eMeeting is an application for synchronous collaboration, already existing (legacy) that has been partially adapted to use the Grid4All API, based initially on a pure centralized structure with lightweight clients, developed on a commercial environment (Flash + FMS). This online collaboration tool permits users to share not only the voice and video but documents, charting and polling and can be seen as a perfect companion for the rest of the Grid4All applications. It can be inserted-into or merged-with any other application giving a solution for the "communication part" of any collaborative session.

CFS is an application for asynchronous collaboration (provides workspaces for file sharing and discussion) developed in the well-known Firefox environment. Concurrent and disconnected modifications to the shared workspaces are managed in a decentralized manner using Telex.

The **Sakura Shared Calendar** co-operative calendar application is implemented over Telex. This application enables any user to share her calendar, to create and invite users to a meeting, to change its time, or to cancel it. Sakura maintains the invariant that no user is double-booked into two different meetings at the same time. In contrast to common calendar systems such as Doodle, Sakura, using Telex ensures consistency even when a user is tentatively engaged in several meetings, i.e., the agreement protocol will not commit conflicting meetings.

CNSE is an application to carry out collaborative network simulations for learning purposes. In this sense, the current prototype of CNSE adds the following features, compared to other network simulators: Collaborative Visualization, Group Awareness and Group Management. CNSE is a comprehensive application that is Grid-aware. Number of open and virtual universities and remote learning practices using simulation to support learning by synthetic experimentation is growing. CNSE provide an integrated environment where groups of students learn by performing collaborative simulation. These methods are computationally intensive. Self-management capabilities will increase adoption of CNSE so that it can be deployed on dynamic, volatile environment such as those envisaged by the project. Niche middleware and the YACS (*Yet Another Computing Service*) framework are appropriate technologies for CNSE to attain this.

We had also showed how deployment of compute intensive services could exploit Grid4All technologies. **gMovie**, is an application that transcodes videos. It may be used for domestic multimedia processing. It may also be used to deploy such transcoders as services. It allows people to access via a web interface a video trans-coding service using distributed processing power. It uses VOFS to distribute input video chunks and collect output, uses ADL-based tools to deploy legacy middleware. It uses a scheduler (developed for Grid4All) in an offline mode to compute worst-case execution times. Such estimations are used to plan capacities and reserve computational resources.

CFS, CNSE and SC are being developed specifically for the project and EM is a legacy application that is adapted and extended to use and benefit from the Grid4All middleware. These applications provide end-user support for different collaborative tasks. Concurrent sharing and modification of documents is enabled by Telex. VOFS is used to provide VO-wide shared persistence services. While CNSE has been re-designed to use Niche and YACS, the integration is not complete.

1.3.6 Grid4All in the light of the state-of-the-art

The Grid4All FP6 project was started in 2006 with the goal of a democratic grid. Since then, Cloud Computing has become all the rage. Providers such as Amazon or Google offer vast amounts of CPU, storage and networking capacities. They are available over the Internet, through very simple interfaces, at small incremental cost. Users can outsource complex tasks, such as system administration, or failure detection and recovery. So, has cloud computing achieved our goal? Has Grid4All been rendered obsolete?

Cloud computing promises easy access to scalable, virtualised resources, billed on a utility model, over a simple internet connection. Resources are diverse, and may include computing power, storage capacity, data, or software. Current clouds are proprietary. They are owned and managed by a commercial provider (e.g., Amazon or IBM), who manages the infrastructure; the user cannot control it and may ignore operational details.

Nevertheless, beyond the hype, something substantial is happening. Clouds allow easy access to plentiful resources, in line with the Grid4All objectives. Cloud computing may be technically quite similar to grid computing, but there is also something more; for instance, an order-of-magnitude increase in scale, which creates many new problems.

Different levels of functionality are offered, such as Infrastructure-as-a-service (IaaS CPU, network and disk), Platform-as-a-Service (PaaS development), Software-as-a-Service (SaaS applications), etc. Clouds bring existing distributed computing concepts to a very large scale by relying on cluster architectures, virtualisation, highly automated management, and extensive failure detection and recovery. From the user's perspective, advantages of cloud computing include paying only for resources effectively used; low barrier to entry; outsourced management; immediate access to virtually unlimited resources; reliability and availability; simple environments.

As for providers, they benefit from a consolidated infrastructure and shared costs, which help create new services and capture more customers. Although some providers currently specialise in renting infrastructure (e.g., IBM, HP, Sun, Dell, Akamai, or VMWare), and others in specific services (e.g., Amazon, Google, Yahoo, Microsoft, Salesforce, or Netsuite), these two models will likely soon converge.

However, this is not the whole story. Current cloud architectures are based on big data centres owned by a single vendor and provide stripped-down functionality (e.g., mostly stateless services). They are proprietary and centralised. This introduces a single organisational point of failure, restricts the kind of applications that are supported, causes vendor lock-in, and deprives users of control over their own data. This has some advantages, but it is not the only possible model. Many efforts are underway to build so-called volunteer clouds, or edge clouds, building on P2P techniques to federate available edge computing resources in a cooperative, tit-for-tat manner.

Researchers in distributed systems are already busy working on increasing scalability even further, while finding remedies to the drawbacks listed above. In particular, we expect future clouds to be considerably more open and less limited in functionality. Standardised interfaces will allow integration of the users own infrastructure, proprietary service providers, and pooled resources, forming a hybrid cloud. Such hybrid clouds will combine edge resources and proprietary clouds, as well as bridging data-centre clouds with one another. In addition to the current single-user applications, the hybrid cloud will offer support for collaboration, data consistency, forming VOs that can aggregate all kinds of resources, support for general-purpose application management, more user control, etc.

Thus, the future points to a hybrid cloud, the seamless integration of a number of providers of various sizes (including the users themselves), offering a vast, dynamic array of services and resources, under largely variable conditions and SLAs. The hybrid cloud will be large in scale and subject to highly variable latencies, churn rates, and frequent disconnection and failure events. Consistent sharing of stateful data will be a requirement.

These are precisely the focus areas of the Grid4All project. Grid4All enables open, democratic access to computing resources, supporting both grid and edge resources, and supports co-operation within distributed groups of users. As a research project, Grid4All focused on advanced technological enablers: peer-to-peer based autonomic management facilities, resource market, and federative data storage and access (and ignored other topics that are either well established or out of its scope). We argue that the Grid4All achievements are enablers for future hybrid clouds. On these crucial points, Grid4All proposes a number of contributions, surveyed in section 2. We believe that technological and economic forces will cause these barriers to be shattered, in the space of a few years. Smaller, more specialised providers will appear. Users will want higher levels of service (for instance, consistency guarantees for their data), will need to access a multiplicity of providers. Some classes of users will prefer to swap resources with one another rather than pay or trust a provider, except possibly for occasional needs.

1.4 Impact of the project on industry and research sector

The project objectives have been accomplished by achieving tangible results that are described in this section. Most of the software results generated by the project are provided through Open Source Licences. These are available at specific websites from where both the software and documentation can be downloaded.

1.4.1 Publishable exploitable knowledge

The following table lists the publishable results and the consortium members that own the result:

Result	Owner(s)	Description
Niche (DCMS)	KTH, SICS, INRIA	A general purpose distributed component management system (DCMS) that includes a component programming model, a corresponding API, tools and ADL interpreter) and an execution environment for development, deployment and execution of distributed self-managing component-based systems, servers and applications.
SIS	UPRC	Semantic discovery and selection service. Specific support for selection of auction markets for computation resources.
GRIMP	UPC, FT	Tools for open market place including configurable auction server framework and mechanisms; decentralized overlay-based market information service and decentralized account management system.
Telex	INRIA	Middleware to write document-based collaborative applications
P2P-LTR	INRIA	P2P log and time-stamping for reconciliation for collaborative applications.
SbQA	INRIA	Satisfaction based query allocation service used as selection service of SIS
VOFS	ICCS	User-centric peer-to-peer file sharing service
Sakura SC	INRIA	Shared calendar using Telex
CFS	UPC	Collaborative File Sharing application
CNSE	UPC	Collaborative Network Simulation Environment
eMeeting	Antares	Tools for collaboration

The complete set of exploitable knowledge is presented in section 2.

1.4.2 Publications at scientific meetings and conferences

Grid4All publications over the entire project duration are presented in the following table:

Publication Title	Authors	Journal	Su/Pu[4]	Date
P2P Technologies for Emerging Wide-Area Collaborative Services and Applications	UPC	Submitted for publication in journal Computer Networks, Special Issue	Pu	2009 (Review notification 15 Nov. 2009)
About Atlas (service selection)	INRIA	VLDB Journal 2009	Pu	To be announced
Churn Tolerant Virtual Organization File System for Grids –	L. Lindback, V. Vlassov, S. Mokarizadeh, and G, Violino,	Proceedings of the accepted for the 4th Grid Applications and Middleware Workshop (GAMW'2009), Wroclaw, Poland, September 13-16, 2009	Pu	To be announced
A Design Methodology for Self-Management in Distributed Environments – a	A. Al-Shishtawy, V. Vlassov, P. Brand, Seif Haridi	Proceedings of the 2009 IEEE International Symposium on Scientific and Engineering Computing (SEC-09), Vancouver, Canada, August 29-31, 2009	Pu	To be announced
The Grid4All ontology for the retrieval of traded resources in a market-oriented Grid	K. Kotis, G. A. Vouros, A. Valarakos, A. Papasalouros (UPRC), X. Vilajosana (UOC), R. Krishnaswamy, N. Amara-Hachmi (FT)	Proceedings of the Workshops published by IEEE Computer Society Press	Pu	To be announced

[4] Su= Submitted, Pu= Published

Publication Title	Authors	Journal	Su/Pu[4]	Date
Paper "Managing democratic grids: architecture and lessons learnt"	Leandro Navarro	Paper at the "Workshop Grids Meet Autonomic Computing", Barcelona, Spain, 15 June 2009. Associated with the 6th International Conference on Autonomic Computing (ICAC'09). Proceedings of the 6th international conference industry session on Grids meets autonomic computing, Barcelona, Spain,	Pu	15 June 2009
Tools for Architecture Based Autonomic Systems,	N. De Palma, K. Popov, N. Parlavantzas, P. Brand, and V. Vlassov,	Proceedings of the Fifth International Conference on Autonomic and Autonomous Systems, 2009. ICAS '09.	Pu	20-25 April 2009
Distributed Control Loop Patterns for Managing Distributed Applications,	A. Al-Shishtawy, J. Höglund, K. Popov, N. Parlavantzas, V. Vlassov, and P. Brand,	Proceedings of the 2008 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops,	Pu	October 2008
A Democratic Grid: Collaboration, Sharing and Computing for Everyone, In: Collaboration and the Knowledge Economy: Issues, Applications, Case Studies, Paul Cunningham and Miriam Cunningham	R. Krishnaswami, L., and V. Vlassov,	(Eds), IOS Press, 2008 Amsterdam, ISBN 978-1-58603-924-0	Pu	October 2008
A Self-Adaptable Query Allocation (demonstration paper)	J. Quiané-Ruiz, P. Lamarre, P. Valduriez	To appear in the Journées Francophones sur les Bases de Données Avancées (BDA).	Su	October 2008

Publication Title	Authors	Journal	Su/Pu[4]	Date
Fault-Tolerant Partial Replication in Large-Scale Database Systems	P. Sutra, M. Shapiro	Euro-Par 2008, Las Palmas de Gran Canaria, Spain,	Su	August 2008
Enabling Self-Management of Component Based Distributed Applications	A. Al-Shishtawy, J. Hoglund, K. Popov, N. Parlavantzas, V. Vlassov and P. Brand	CoreGRID Symposium 2008, August 25-26, 2008, Las Palmas de Gran Canaria, Canary Island, Spain	Su	August 2008
Autonomic Management of Distributed Systems and Applications Using Structured Overlay Networks	A. Al-Shishtawy, J. Hoglund, K. Popov, N. Parlavantzas, V. Vlassov and P. Brand	The 12-th International Conference On Principles Of Distributed Systems (OPODIS'08), Dec 2008, Luxor, Egypt	Su	July 2008
Design and Implementation of a Virtual Organization File System for Dynamic VOs	H.-R. Mizani, L. Zheng, V. Vlassov, K. Popov	International Symposium on Scientific and Engineering Computing (SEC-08), Sao Paulo, Brazil, July 16-18, 2008 – to appear	Pu	July 2008
Telex: Principled System Support for Write-Sharing in Collaborative Applications	L. Benmouffok, J-M. Busca, J. M. Marquès, M. Shapiro, P. Sutra, G. Tsoukalas.	Rapport de recherche INRIA RR-6546	Pu	May 2008
Towards decentralized resource allocation for collaborative peer to peer learning environments	X. Vilajosana, D. Lázaro , A. A. Juan, L. Navarro.	CESA2008: International Workshop on Collaborative e-Learning Systems and Applications held in conjunction with CISIS -2008 International conference. IEEE Proceedings.	Su	4-7 March 2008
Bidding support for computational resources	X. Vilajosana, J-M. Marques, R. Krishnaswamy, A. A. Juan, N. Amara-Hachmi, L. Navarro	The International Workshop on P2P, Parallel, Grid and Internet Computing (3PGIC-2008) http://www.ares-conference.eu/3PIG-2008/ IEEE Computer Society.	Su	4 March 2008
Managing Virtual Money for	J. Quiané-Ruiz,	In Proceedings of the EDBT	Pu	March 2008

Publication Title	Authors	Journal	Su/Pu[4]	Date
Satisfaction and Scale Up in P2P Systems	P. Lamarre, S. Cazalens, P. Valduriez	Workshops: Data Management in Peer-to-Peer Systems (DAMAP)		
Grid4All Press note	G4A Consortium	ESADE Magazine	Su (future)	February 2008
Grid4All Press note	G4A Consortium	Capital Humano	Su (future)	February 2008
BitDew: A Programmable Environment for Large-Scale Data Management and Distribution	G. Fedak, H. He, F. Cappello.	INRIA Research Report 6427, INRIA Saclay – Île-de-France	Pu	January 2008
Special issue on "Web/Grid Information and Services Discovery and Management",	UPRC, UPC, FT/University of the Aegean, Ai-Lab	Int. J. Web and Grid Services, 4(4):418-439	Pu	2008
Handbook of Research on Social Dimensions of Semantic Technologies and Web Services,	M. Manuela Cunha, Eva Oliveira, Antonio Tavares & Luis Ferreira	(Eds), Information Science Reference, ISBN: 978-1-60566-650-1	Pu	2008
Future Generation Computer Systems.	UPRC, INRIA / University of the Aegean, Ai-Lab	The International Journal of Grid Computing and eScience (Elsevier)	Pu	Submitted
Elaborating a Decentralized Market Information System.	R. Brunner, F. Freitag	OTM Workshops (1) 2007: 245-254	Pu	25-30 Nov. 2007
Towards an Open Grid Marketplace Framework for Resources Trade.	N. Amara-Hachmi, X. Vilajosana, R. Krishnaswamy, L. Navarro-Moldes, J-M. Marques	OTM Conferences (2) 2007: 1322-1330	Pu	25-30 Nov. 2007
A Comparison of Optimistic Approaches to Collaborative Editing of Wiki Pages	C.-L. Ignat, G. Oster, P. Molli, M. Cart, J. Ferrié, A.-M. Kermarrec, P. Sutra, M. Shapiro, L. Benmouffok, J.-M. Busca, R. Guerraoui.	CollaborateCom	Pu	November 2007
A Comparison of Optimistic Approaches to Collaborative Editing of Wiki Pages	C.-L. Ignat, G. Oster, P. Molli, M. Cart, J. Ferrié, A.-M. Kermarrec, P. Sutra, M. Shapiro, L. Benmouffok, J.-M. Busca, R. Guerraoui.	Int. Conf. on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), White Plains,NY,USA	Pu	November 2007

Publication Title	Authors	Journal	Su/Pu[4]	Date
Decentralised Commitment for Optimistic Semantic Replication	P. Sutra, J. Barreto, M. Shapiro.	Int. Conf. on Co-operative Information Systems (CoopIS)	Pu	November 2007
Comparing Optimistic Database Replication Techniques	P. Sutra, M. Shapiro.	Bases de données avancées	Pu	October 2007
Designing a commutative replicated data type	M. Shapiro, N. Preguiça.	Rapport de recherche INRIA RR-6320	Pu	October 2007
Global-scale peer-to-peer file services with DFS	A. Chazapis, G. Tsoukalas, G. Verigakis, K. Kourtis, A. Sotiropoulos, N. Koziris	Proceedings of the 8th IEEE/ACM International Conference on Grid Computing (Grid 2007) - ISBN 1-4244-1560-8	Pu	21 Sept. 2007
Semantic Middleware for Designing Collaborative Applications in Mobile Environment	L. Benmouffok, J-M. Busca, M. Shapiro.	MiNEMA Workshop on Middleware for Network Eccentric and Mobile Apps. Magdebourg, Germany	Pu	September 2007
Characterizing Result Errors in Internet Desktop Grids	D. Kondo, F. Araujo, P. Malecot, P. Domingues, L. Silva, G. Fedak, F. Cappello	European Conference on Parallel and Distributed Computing, EuroPar Rennes, best paper award, INRIA Saclay - Île-de-France	Pu	August 2007
Towards Efficient Data Distribution on Computational Desktop Grids with BitTorrent	B. Wei, G. Fedak, F. Cappello	FGCS	Pu	August 2007
The Role of Overlay Services in a Self-Managing Framework for Dynamic Virtual Organizations	P. Brand, J. Hoglund, K. Popov, N. de Palma, F. Boyer, N. Parlvanzas, V.Vlassov, A. Al-Shishtawy	CoreGRID Workshop on Grid Programming Model, Grid and P2P Systems Architecture, Grid Systems, Tools and Environments, Heraklion - Crete, Greece, 2007. To appear in the Springer CoreGRID series.	Pu	June 2007

Publication Title	Authors	Journal	Su/Pu[4]	Date
Exploiting our computational surroundings for better mobile collaboration	J. Barreto, P. Ferreira, M. Shapiro.	Int. Conf. on Mobile Data Management (MDM'07), Mannheim, Germany	Pu	May 2007
A Flexible Mediation Process for Large Distributed Information Systems	P. Lamarre, S. Lemp, S. Cazalens, P. Valduriez	Int. Journal of Cooperative Information Systems, 16(2), 299-332	Pu	2007
A Topology-aware Approach for Distributed Data Reconciliation in P2P Networks	M. El Dick, V. Martins, E. Pacitti.	European Conf. on Parallel Computing (Euro-Par), Rennes	Pu	2007
Data Currency in Replicated DHTs	R. Akbarinia, E. Pacitti, P. Valduriez.	ACM SIGMOD Int. Conf. on Management of Data,/Beijing, China, 211-222.	Pu	2007
Data management in the APPA P2P system	R. Akbarinia, V. Martins.	Journal of Grid Computing, 5(3)	Pu	2007
Design of PeerSum: A Summary Service for P2P Applications	R. Hayek, G. Raschia, P. Valduriez, N. Mouaddib.	Int. Conf. on Advances in Grid and Pervasive Computing (GPC),/ Paris, LNCS 4459 Springer 2007, 13-26.	Pu	2007
Grid Data Management: open problems and new issues	E. Pacitti, P. Valduriez, M. Mattoso.	Journal of Grid Computing, 5(3), 273-281	Pu	2007
KnBest - A Balanced Request Allocation Method for Distributed Information.	J. Quiané-Ruiz, P. Lamarre, P. Valduriez.	Int. Conf. on Database Systems for Advanced Applications (DASFAA),/ Bangkok, Thailand, 237-248	Pu	2007
Position Algorithms for Top-k Queries	R. Akbarinia, E. Pacitti, P. Valduriez. Best	Int. Conf. on Very Large Databases (VLDB), Vienna, Austria	Pu	2007
Processing Top-k Queries in Distributed Hash Tables	R. Akbarinia, E. Pacitti, P. Valduriez.	European Conf. on Parallel Computing (Euro-Par)/, Rennes,	Pu	2007

Publication Title	Authors	Journal	Su/Pu[4]	Date
SQLB: A Query Allocation Framework for Autonomous Consumers and Providers	J. Quiané-Ruiz, P. Lamarre, P. Valduriez.	Int. Conf. on Very Large Databases (VLDB),/ Vienna, Austria,	Pu	2007
Top-k Query Processing in the APPA P2P System	R. Akbarinia, V. Martins, E. Pacitti, P. Valduriez.	Int. Conf. on High Performance Computing for Computational Science (VecPar 2006)/, LNCS 4395, Springer, 158-171	Pu	2007
DyMRA: Dynamic Market Deployment for Decentralized Resource Allocation.	D. Lázaro, X. Vilajosana, J-M. Marquès	OTM Workshops (1) 2007: 53-63	Pu	2007
On Resource Volatility in Enterprise Desktop Grids	D. Kondo, G. Fedak, F. Cappello, A.A. Chien, H. Casanova	2nd IEEE International Conference on e-Science and Grid Computing (eScience'06)	Pu	Dec. 2006
XtremLab: Une plateforme pour l'observation et la caractérisation des grilles de PC sur Internet	P. Malécot, D. Kondo, G. Fedak	Rencontres francophones du parallélisme (Renpar'06)	Pu	Oct. 2006
XtremLab: A System for Characterizing Internet Desktop Grids	P. Malécot, D. Kondo, G. Fedak	Poster in International Symposium on High Performance Distributed Computing HPDC'06	Pu	June 2006
The Computational and Storage Potential of Volunteer Computing	D.P. Anderson, G. Fedak	6th International Symposium on Cluster Computing and the Grid (CCGRID'06)	Pu	May 2006
Towards Soft Real-Time Applications on Enterprise Desktop Grids	D. Kondo, B. Kindarji, G. Fedak, F. Cappello	6th IEEE Symposium on Cluster Computing and the Grid (CCGRID '06)	Pu	May 2006

Publication Title	Authors	Journal	Su/Pu[4]	Date
Reconciliation in the APPA P2P System	V. Martins, R. Akbarinia, E. Pacitti, P. Valduriez.	IEEE Int. Conf. on Parallel and Distributed Systems (ICPADS)/, Minneapolis, USA, 401-410	Pu	2006
Reducing Network Traffic in Unstructured P2P Systems Using Top-K Queries	R. Akbarinia, E. Pacitti, P. Valduriez	Distributed and Parallel Databases, 19(2), 67-86	Pu	2006
Reducing Network Traffic in Unstructured P2P Systems Using Top-K Queries	R. Akbarinia, E. Pacitti, P. Valduriez.	Distributed and Parallel Databases/, Kluwer Academic Publishers, 19(2), 67-86	Pu	2006
Resource Availability in Enterprise Desktop Grids,	D. Kondo, G. Fedak, F. Cappello, A;A.Chien, H. Casanova	Future Generation Computer Systems	Pu	2006
Satisfaction Based Query Load Balancing	J. Quiané-Ruiz, P. Lamarre, P. Valduriez.	Int. Conf. on Cooperative Information Systems (CoopIS'06), /Montpellier, 36-53	Pu	2006

Major Grid4All dissemination activities in 2008 and 2009 are presented in the following table:

Planned /actual Dates	**Major dissemination activity (description)**	**Relevant details**
July 16-18, 2008	Presentation at a conference and publication in conference proceedings	H.-R. Mizani, L. Zheng, V. Vlassov, and K. Popov, Design and Implementation of a Virtual Organization File System for Dynamic VOs, In: Computational Science and Engineering Workshops, 2008. CSEWORKSHOPS '08. 11th IEEE International Conference on, IEEE Computer Society Press, pp.77-82, July 2008
August 25-26, 2008	Presentation at a conference and publication in conference proceedings	A. Al-Shishtawy, J. Hoglund, K. Popov, Nikos P., V. Vlassov, and P. Brand, Enabling Self-Management of Component Based Distributed Applications, In: From Grids to Service and Pervasive Computing. Proc. of the CoreGRID symposium 2008, Las Palmas de Gran Canaria, Spain, August 2008. pp. 163-174, CoreGRID Series. Springer.
October 21, 2008	Presentation at a conference and publication in conference proceedings	A. Al-Shishtawy, J. Höglund, K. Popov, N. Parlavantzas, V. Vlassov, and P. Brand, Distributed Control Loop Patterns for Managing Distributed Applications, In: 2008 2-nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, 2008, IEEE Computer Society, pp. 260-265, Oct. 2008

Planned /actual Dates	Major dissemination activity (description)	Relevant details
October 2008	Presentation at a conference and publication in a book	R. Krishnaswami, L. Navarro, and V. Vlassov, A Democratic Grid: Collaboration, Sharing and Computing for Everyone, In: Collaboration and the Knowledge Economy: Issues, Applications, Case Studies, Paul Cunningham and Miriam Cunningham (Eds), IOS Press, 2008 Amsterdam, ISBN 978-1-58603-924-0
July 1-2, 2008	Presentation "VO management in Grid4All" by Vladimir Vlassov, KTH	Presentation at the joint Grid4All-XtreemOS technical meeting, July 1-2, 2008, LIP6, Paris
July 2-3, 2008	Presentation "VO management in Grid4All" by Vladimir Vlassov, KTH	Presentation at the CT1 TG8 VO meeting, July 2008, LIP6, Paris
April 20-25, 2009	Presentation at a conference and publication in conference proceedings	N. De Palma, K. Popov, N. Parlavantzas, P. Brand, and V. Vlassov, Tools for Architecture Based Autonomic Systems, Fifth International Conference on Autonomic and Autonomous Systems, 2009. ICAS '09. 20-25 April 2009 Page(s):313 – 320
June 11, 2009	Presentation "Grid4All: Results and Lessons Learnt" by Vladimir Vlassov, KTH	Presentation at the CT1 TG8 VO meeting, June 11, 2009, Brussels
June 10-11, 2009	Poster "Grid4All: Results and Lessons Learnt" presented by Vladimir Vlassov, KTH	Poster at the Collaboration Meeting for FP6 & FP7 projects, Internet of Services 2009, 10-11 June 2009
June 15 2009	Paper "Managing democratic grids: architecture and lessons learnt"	Paper at the "Workshop Grids Meet Autonomic Computing", Barcelona, Spain, 15 June 2009. Associated with the 6th International Conference on Autonomic Computing (ICAC'09). Proceedings of the 6th international conference industry session on Grids meets autonomic computing, Barcelona, Spain, Pages 51-52, 2009, ISBN:978-1-60558-578-9
August 29-31, 2009	Presentation at a conference and publication in conference proceedings	A. Al-Shishtawy, V. Vlassov, P. Brand, Seif Haridi, A Design Methodology for Self-Management in Distributed Environments – accepted for the 2009 IEEE International Symposium on Scientific and Engineering Computing (SEC-09), Vancouver, Canada, August 29-31, 2009
September 13-16, 2009	Presentation at a conference and publication in conference proceedings	L. Lindback, V. Vlassov, S. Mokarizadeh, and G, Violino, Churn Tolerant Virtual Organization File System for Grids – accepted for the 4th Grid Applications and Middleware Workshop (GAMW'2009), Wroclaw, Poland, September 13-16, 2009
June 2-3 2009	Demonstration of market-place tools at the Orange-wide research salon.	A web-based portal application showing the use of auction mechanisms to trade computational resources was demonstrated at the Orange-wide research salon. This demonstration was one out five such demonstrations short-listed for VIP visitors (including Didier Lombard and other members of the Orange general management committee and also senior managers within Orange Labs and other business units). The feedback was rather favorable. Signals received were (a) continue both research and development work in this dimension in perspective of mediation and market places for Clouds.

Planned /actual Dates	Major dissemination activity (description)	Relevant details
Planned for 2009	Chapter in "Economic Models and Algorithms for Distributed Systems" to be published by Birkhauser, a Springer affiliate.	Work conducted within FT on combinatorial auction mechanisms within the scope of democratizing Grid technologies, has been submitted for publication. This book is edited by Dirk Neumann, Chair for Information Systems Research at Alber-Ludwigs-Universitat Frieiburg, Germany

1.4.3 Dissemination to decision makers and general audience

The Grid4All consortium publishes its activities to a diverse audience. These activities are concentrated in the project website, project reports and newsletters, participation at exhibitions, and presentations to industry.

INRIA (Regal) organized a workshop in May 2009 (called IliaTech https://www-c.inria.fr/Internet/rendez-vous/iliatech/pair-pair). This workshop was open to academia and industry. The objective of this technology day was to present outstanding results in domain of collaboration-tools and social networking. Seminars on Telex and demonstrations of Telex-based applications were presented at the workshop. The demo presented the Telex-based shared calendar application, Sakura-SC, which now features the Sunbird GUI. Various scenarios involving three users were presented, including complex conflicts on grouped events. For each independent conflict, Sakura-SC displayed the solutions to the conflict and users were able to select and commit a particular solution.

France Telecom (Orange Labs) conducts an Orange-wide research salon twice every year. Cloud computing was the showpiece of 2009's fair. Technological capabilities to federate different data centers and aggregate their capacities and democratization of supply side of cloud computing are believed to be necessary to achieve the full promises of Cloud computing. This not only stimulates competition but also avoids longer-run issues such as vendor-lock in. We demonstrated tools developed within the project to show how Cloud market places may be deployed and operated. Through a portal based application, the utility of combinatorial auction mechanisms to aggregate computing capacity and the practical value of obtaining compatible linear prices were shown. The feedback was in general positive; it raised awareness and debates on how market-based tools could be exploited for Cloud brokering.

SICS conducts an Open House day every year. SICS and KTH presented a poster and a demonstration of gMovie. This demonstration consisted of a transcoding application that was designed to use YACS (Yet Another Computing Service) which itself is a self-managing Niche service.

UPC and KTH jointly submitted a joint Master program proposal to the Erasmus Mundus program in April 2009. This European Master in Distributed Computing proposal, recently approved, is a natural domain of application for the Grid4All results.

While this section has listed activities conducted during the lifetime of the project, clearly dissemination and exploitation activities will contribute beyond the project life. Some steps to be followed after the end of the project:

- Promotion of project results by the organisations involved in the project. The individual exploitation plans of each partner show how each organisation plans to promote and/or use Grid4All results.
- Reuse of knowledge and results in future collaborative projects.
- Reuse of knowledge and results in academia (courses, Master and PhD programs).
- Seminars and tutorials on the software results.
- Maintenance of the project Web site.
- Hosting of software results at popular web sites.

1.5 Lessons learnt and future work

While we believe that the project has done significant progress towards achieving its goals, the following insights have been obtained:

- A middleware, such as Niche, clearly reduces burden from an application developer, because it enables and supports self-management by leveraging self-organizing properties of structured P2P overlays and by providing useful overlay services such as deployment, DHT (can be used for different indexes) and namebased communication. However, it comes at a cost of self-management overhead, in particular, the cost of monitoring and replication of management; though this cost is necessary for the democratic grid (or cloud) that operates on a dynamic environment and requires self-management.

  There is the issue of coupled control loops, which we did not study. In our scenario multiple managers are directly or indirectly (via stigmergy) interacting with each other and it is not always clear how to avoid undesirable behaviour such as rapid or large oscillations which not only can cause the system to behave nonoptimally but also increase management overhead. We found, as mentioned above, that it is desirable to decentralize management as much as possible, but this probably aggravates the problems with coupled control loops. Every application (or service) programmer should not need to handle co-ordination of multiple managers (where each manager may be responsible for a specific behaviour). Future work should address design of coordination protocols that could be directly used or specialized.

  Although some overhead of monitoring for self-management is unavoidable, there are opportunities for research on efficient monitoring and information gathering/aggregating infrastructures to reduce this overhead. While performance is not perhaps the dominant concern of 'democratic grid' users, we believe that this should be a focus point since monitoring infrastructure itself executes on volatile resources.

  Replication of management elements is a general way to achieve robustness of self-management, especially, self-healing. In fact, most evaluators assumed that management programs will be robust. They did not build-in protection from failure of management logic. Even though we have developed and validated a solution (including distributed algorithms) for replication of management elements in Niche, it is reasonable to continue research on efficient management replication mechanisms.

- A major concern that arises is ease of programming of management logic. Research should focus on high-level programming abstractions, language support and tools that facilitate development of selfmanaging applications. We have already started to address this aspect. The management tools such as DepOZ reduce the complexity of deploying and reconfiguring applications running on large-scale, dynamic environments. The DepOz programming framework provides flexible support for configuration and dynamic reconfiguration. It includes support for navigating and monitoring component structures as well as for defining workflows in a concise and compositional way.

  While DepOz currently provides high flexibility without sacrificing ease of use, the next step should be directed towards deployment on large-scale systems. Optimization, of both the deployment process and resource usage is complex. DepOz could exploit research using techniques such as constraint-based programming to address this.

- While VOFS is a simple but flexible collaboration tool, allowing peers to share data through linking and federations, supports disconnected collaboration modes and importantly does not require expertise to setup and use, future avenues to explore could include support for better resource (storage) and replica control and better integration with Telex to support concurrent operations.

- Telex helps developers to build collaborative applications faster. It supports an optimistic replication model for sharing stateful data in a decentralised way over a large-scale network. Telex eases application development by taking care of the communication, replication and consistency issues. Based on a principled approach, Telex guaranties that replicas never violate safety and converge eventually. The platform is designed for robustness, flexibility and performance. Experience with applications and benchmarks has given useful feedback, both regarding the implementation of Telex, as well as guidelines for application developers.

The current implementation of Telex suffers from excessive memory consumption. The ACG can quickly reach sizes of several tens of thousands of nodes, and is accessed concurrently by many threads. For instance, the scheduler parses the ACG at the same time as local and remote applications are modifying it. To avoid concurrency issues, the scheduler takes a full copy of the current ACG, which both consumes memory and is slow (in Java). Similarly, forward execution and rollback of applications involves copying their internal state, which can be very large. In both cases, an obvious solution (and future work) is to copy-on-write instead.

Constraints are hard to validate. We suggest two complementary approaches for future work. A compiler could generate actions and constraints from a high-level specification, and a checker could verify that all action-constraint combinations verify the application invariants.

- Economic methods for resource allocation are promising. The pricing mechanisms and the heuristics-based algorithms designed to resolve the allocation problem in the combinatorial auction bring trustable information to participants, and provide good guidelines for participants (bidders). Moreover we compute prices as a function of time. This is useful since it helps participants to plan their needed capacity and adapt their requests over time. Though linear prices are very attractive due to their intuitive meaning to the bidder, their calculation requires significant computational effort and some times compatible prices may not exist and we may not always find even good enough approximations. This requires improvements on the proposed scheme such that prices can be correctly interpreted.

  Learning technology evolutions is difficult to automate. An open market-place may employ a wide category of pricing-based or even other multi-criteria based negotiation protocols and mechanisms. Actors in the market place may take different roles, e.g. buyers, sellers, auctioneers. These agents may not be doted with all possible negotiation capabilities, either for economic purposes or for interaction-behaviour purposes. We need to minimally allow participants to understand their capability with respect to a given auction protocol and possibly allow the participant to adapt to behaviours of the auctioneer. We advocate an open service oriented market place where negotiation protocols may be formalized using choreography description languages such as WS-CDL. It is interesting to pursue work in this direction to detect compatibility, i.e. is a trader doted with the behaviour expected of the auction at which it wants to participate and then to find mechanisms that permit traders to adapt.

- The G4A-SIS offers programmatic G4A-SIS extends current discovery services by permitting market-oriented characterization of resources to match offers and request. Preferences of both consumers and providers are used in this selection process. Providers and consumers may specify composite resources, i.e. bundles of multiple types of resources. This feature simplifies allocation of complex resource requests.

  Even though the SIS offers Java-based programmatic interfaces, this API is dependent on the underlying ontology for market descriptions. Modifications to this ontology imply changes to the API. The selection service matches consumers and providers such that their preferences are mutually satisfied. Preferences may involve dynamic parameters e.g. provider loads. Such deployments will require a scalable architecture to monitor and update preference values. The current design has a centralized architecture; this limits scalability, restricts autonomy and breaks the desire for openness. We have finalized the design of a decentralized architecture that hopefully will remove these restrictions.

1.6 Project logo

1.7 Project web site

1.7.1 Public website

The Grid4All public website is available at the following link: http://www.grid4all.eu

The original website has been totally reorganised and redesigned. The new interface has been made available in January 2008. ANTARES is responsible for the regular updates and maintenance of the website.

1.7.2 Grid4All newsletter

The 4th newsletter published at the project web site is presented below.

Grid4All democratize the grid

NEWSLetter

issue n° 4, March 2009

Events

The SICS Open House
Stockholm, Sweden, 24 April 2009

The SICS Open House is a opportunity for both industry and the general public to take part of the research done at SICS and interact with the researchers who present their works through talks and demonstrations.

We illustrate our distributed component management system, DCMS, for self-managing application by GMovie. GMovie is a CPU-intensive video trans-coding application that shortens execution time by splitting and distributing subtasks to multiple compute resources. With DCMS support GMovie performs self-management by healing when resources fail and configuring by adding/removing resources on demand.

Workshop Grids Meet Autonomic Computing
Barcelona, Spain, 15 June 2009
Associated with the 6th International Conference on Autonomic Computing (ICAC'09)

Managing democratic grids: architecture and lessons learnt by Leandro Navarro (UPC)

The talk will present the architecture, results and lessons learned in the EU project Grid4All (2006-2009) on the construction of a democratic grid infrastructure, where self-* properties are an essential element.

These grids have challenging and novel requirements: usability, content sharing and collaboration, security, scalability and availability via decentralization, brokerage of resources, and autonomic management of the dynamics of organizations and applications. These are considered essential factors for next generation Internet computing and generic enablers for cloud computing.

The talk will present the architecture of the Grid4All infrastructure, the self-* mechanisms incorporated to enable these grids built on conventional computers and networks contributed by the participants, and the lessons learnt during the construction and evaluation.

30 Months work

After 30 Months of work, the Grid4All consortium presents some conclusions obtained from the work done.

Overlay Infrastructure and Programming Models

At the infrastructure and platform level, this team focusses on core VO services, the bottommost layers in the Grid4All architecture. The focus is on **tools to develop services/applications** that can be deployed, run, and manage themselves in dynamic environments characterized by **high rate of churn**.

What's been achieved?

1. Development of the DCMS (Distributed Component Management System). DCMS is a distributed runtime execution system for self-managing applications & services in dynamic VOs. It provides basic support for **deploying distributed applications**, in cooperation with the VO resource service. It supports sensing changes in the state of components and environment, and allows individual components to be found and appropriately manipulated. It deploys management components and sets up the appropriate sensor and actuation support. The current prototype is largely completed; though, replication of management components (for robustness) is still rudimentary. The DCMS is a VO-wide infrastructure based on structured **overlays to provide good scalability and robustness** in the face of churn.
2. Development of a number of tools and abstractions that allow management logic to be specified at a high-level. There is currently an ADL compiler for **describing initial configurations** (which are compiled to DCMS instructions). Other tools, FractOz, Lactoz, demonstrate languages for creating self-configuring components/systems, though they are not yet coupled to the DCMS.
3. Considerable progress toward has been made, i.e. have enabled, the final goal of being able to **offer a full-fledged high-level programming system** for self-managing applications/services in dynamic environments.

As main innovation, we'd like to point the novel use of overlays in the DCMS, where an adapted structured overlay provides a **network-transparent sensing/actuation** infrastructure. We have also adapted known distributed algorithms to dynamic groups. DCMS provides uniform handling of executables and runtime structures as well as parameterized architecture descriptions which support dynamic reconfiguration and reuse of management logic.

Based on this work, the consortium considers that there are many aspects of **scalability** to deal with. For example, the size of the VO itself, the number of applications/services that are supported within it, the number of components and number of individual machines running a single application/service or the number of components in a component group. Without clear end-user-application requirements it is very hard to prioritize which scalability that needs the best support.

A middleware (such as DCMS) clearly reduces burden from an application developer because it enables and supports self-management by leveraging self-organizing properties of structured P2P overlays and by providing useful overlay services such as deployment and name-based communication. However it comes at a cost of self-management overhead, in particular, the cost of monitoring; though this cost is necessary for the democratic grid that operates on a dynamic environment and requires self-management. This opens new opportunities for research on efficient monitoring and information gathering/aggregating infrastructures to reduce this overhead. Another research focus is on high-level programming abstractions and a language support that should facilitate development of self-managing applications.

Virtual Organization and Resource Management

At a next level, this package provides services that are required to provision Virtual Organizations. Organizations that may grow or shrink according to activity conducted. Applications may require more resources or may also need to access needed services.

Semantic information services (SIS) provides access to and manipulate information about **markets trading resources and services** in the Grid environment.

Democratization of Grids implies bridging between different and multiple perspectives of agents participating in the Grid environment. Semantic descriptions and matching between different perspectives at the semantic level tends to be mandatory within a democratic environment. The Semantic Information Service **enables this objective**.

The SIS has shown two important steps towards democratization; stakeholders for SIS are end-users and application developers who can use a web-based GUI to interact with the system. Software agents can interact with SIS via the SIS-API. SIS provides **automatic tools to annotate services**, thus relieving developers of obligations to master semantic technologies. This, together with an important collection of algorithms to align different ontologies can **facilitate the seamless migration** of current Grid and service entities to become semantic Grid entities.

Nevertheless this comes with a cost; the initial **prognosis about scalability** is indeed true. An important lesson learnt is that future research should search a balance between semantic approaches and distributed data management. Work remains to further develop decentralized and distributed services for the manipulation of semantic entities. Future and long term deployment of semantic information services will indeed become a reality.

SIS and Grid4All market place tools are symbiotic. Agents (software/human) discover trading instances for resources and services by querying the semantic information service (SIS). This **provides a ranked list of markets or services** that satisfy user requirements and preferences. Further work will investigate the live update of advertisements' status as the market dynamically evolves over time."

Grid4All approach to decentralized market places relies on efficient access to market feedback that aggregates per commodity prices decided at multiple and simultaneous but independent resource auctions. A number of interesting problems are opened; how to guide tradeoff between accuracy of such information with respect to cost of obtaining such information, how to monitor activity and what intelligent tools may be deployed by market place operators to facilitate agents that require or offer resources.

Data Storage

The goal of the Data Storage is to **develop new technologies** for data sharing, enabling new styles of cooperative work within virtual organisations:

- VOFS, an easy-to-use approach to sharing files by federating users' resources.
- The Telex replication middleware that enables users to edit shared documents and reconcile divergent updates.
- Two systems for concurrent editing, P2P-Xwiki and Treedoc.

These technologies have been **successfully** designed, developed, integrated, demonstrated and evaluated in WP3. Key to the success was clear, **simple and common architectural vision**, resulting from frequent meetings and a lot of joint engineering work.

Interesting research sprung up as an unplanned side-effect of the collaboration: the development of concurrent editing algorithms, work on security in distributed collaborative systems, and research on partial replication.

Distributed, replicated, stateful systems are extremely complex; we learned that, instead of reacting to problems, it is **better to design the system to be immune** to them. For instance, designing concurrent operations to commute avoids concurrency control and reconciliation issues. Another example is the P2P paradigm that automatically converges despite failures and churn.

Scalability remains an open issue. Scalability in number of users is less critical than initially thought, as a collaborative workgroup will rarely be very large. We were unable to find the "killer app" for complex reconciliation scenarios.

The platforms and technologies developed in WP3 will continue to support future research in new areas such as cloud computing and social networks, with a focus on P2P technologies, large-scale data sharing, and privacy.

Applications

This group has been developing or adapting pre-existent applications for a decentralized environment as Grid4All.

Developing applications for the grid4All environment has proven to be more **challenging** than for an idealized (centralized, reliable and stable) model. Partly this difficulty has come from the lack of equivalent tools to facilitate the work under this environment (something already expected). Discounting this, it continues to be harder as application developers have to deal with issues that previously were invisible (transparent) to them. This extra price was also expected, but slightly underestimated. Partly is obviously due to the lack of maturity of the API.

However this extra effort has shown valid when **the benefits of enabling applications to work on a wider range of environmental conditions** are desired (more dynamic and therefore more realistic: e.g. dealing with environmental changes using DCMS, VOMS, GRIMP, indirectly using the Niche overlay), taking advantage of some characteristics of the infrastructure without having to deal with all their drawbacks (Virtual Organizations and VOMS as a way to isolate from boundaries, the Scheduling service and GRIMP to profit from additional resources).

The grid4All infrastructure also **enables applications to share information** in ways that other applications try to avoid (e.g. concurrent modification of shared objects with Telex) leaving the complex details to the infrastructure, and provides a common environment where data (file naming and sharing with VOMS) and complex information (e.g. semantic information about resources with SIS) can be **easily shared across multiple applications** with common security checks.

More information on overall results can be found at www.grid4all.eu.

2. Publishable results

This section provides a per-result table of the complete set of exploitable knowledge produced by Grid4All.

Result Name	**Niche – a Distributed Component Management Service (DCMS)**
Result Owner	SICS, KTH, INRIA Sardes
Sector(s) of application	A wide category of large-scale distributed applications or services that may execute in all kinds of environments, including very dynamic ones with volatile resources.
Direct Stake holder	Developers and service providers
Brief description	Niche is a general-purpose distributed component management system (DCMS) used to develop, deploy and execute self-managing applications or services in all kinds of environments, including dynamic ones with volatile resources. Niche is both a component-based programming model that includes management aspects as well as a VO-wide distributed run-time environment. The Niche platform provides a Java API to develop self-managing applications. Niche API allows a developer to separate functional code from management code so that self-management can be configurable, e.g. enabled or disabled. The management code is build of management elements that include sensors, watchers, and aggregators for monitoring, managers and actuators for making management decisions and executing management actions. Niche API includes basic classes and interfaces for above management elements. Niche runtime environment supports replication of management elements in order to improve robustness of management. Niche API allows the developer to develop a functional part of her application using Niche or to wrap an existing application using Niche API to make the application manageable and to enable its self-management. Niche runtime environment is a set of Niche containers running on computers or VMs. Niche containers form a structured overlay network and provide runtime support for application components, both, management and functional. A self-managing application uses the Niche API to monitor and manage itself, e.g. reconfigure, adapt to changing environment, repair after failures, scale. Underlying Niche overlay network is self-managed; in particular, it transparently corrects routing tables on node failures and leaves, as well as informs application management components subscribed to be notified on those events. In order to demonstrate Niche, two self-managing services have been developed: YASS: Yet Another Storage Service; and YACS, Yet Another Computing Service. Each of the services has self-healing, self-turning and self-configuration capabilities. Each of services implements relatively simple self-management algorithms, which can be changed to be more sophisticated, while reusing existing sensing and actuation code of the services. Both services are included in Niche distribution. Niche (including above demonstrator applications and documentation) is available at http://niche.sics.se
Stage of development	Laboratory prototype
Collaboration sought or offered	Information exchange, application and service developers who are willing to use Niche as a development platform. Platform for course work (laboratory).
Collaborator details	Universities, software developers and vendors who develop large-scale applications requiring self-management capabilities.

Result Name	Niche – a Distributed Component Management Service (DCMS)
IPR granted or published	The 3-clause BSD license
Contact details	Vladimir Vlassov (vladv@kth.se)

Result Name	Telex
Result Owner	INRIA
Sector(s) of application	Collaborative tools such as wikis, decision tools, shared editors, group management etc.
Direct Stake holder	Application developers
Brief description	Telex is a principled system that enables distributed VO users to collaborate over documents (e.g., files, databases, calendars, spreadsheets, text documents, etc.) without prior synchronization. Actions of different users may conflict; Telex resolves conflicts according to semantic parameters provided by Telex-enabled applications. This approach eases the design of collaborative applications, by taking care of system aspects such as replication, conflict repair, eventual commitment and consistency. A collaborative document is shared through a VOFS workspace and has a VO-wide name, is persistent, can be accessed offline, and is subject to mandatory access control. Telex supports multiple-document updates, detects and repairs conflicts between updates by different users, supports peer-to-peer co-operative commitment, and guarantees consistency. Telex may be re-used, as is, outside the Grid4All project.
Stage of development	Laboratory prototype
Collaboration sought or offered	Development of collaborative tools, applications and services using Telex.
Collaborator details	Software vendors of collaborative tools and applications
IPR granted or published	BSD type licence
Contact details	Marc Shapiro INRIA – Regal group (marc.shapiro@acm.org)

Result Name	**VOFS**
Result Owner	ICCS
Sector(s) of application	Shared workspaces with files and storage for both end users and applications. This can be used to implement collaborative applications.
Direct Stake holder	End-user, developer, service provider
Brief description	VOFS is a user-centric peer-to-peer file-system that allows people and applications to create and share files and storage through the well-known file-system interface. VOFS will be exploitable beyond the scope of Grid4All. Its extensibility will enable easy adaptation to specific future scenarios, either Grid4All-derived or not. Adaptation may be needed either by providing different implementations of some pluggable components or inserting policies in core mechanisms.
Stage of development	Laboratory prototype
Collaboration sought or offered	Enhancement of VOFS (robustness, feature set)
Collaborator details	Software developers and vendors in field of distributed file sharing systems
IPR granted or published	Simplified BSD/MIT
Contact details	Georgios Tsoukalas (gtsouk@cslab.ece.ntua.gr)

Result Name	**WebDAV Virtual Organization File System, WebDAV VOFS**
Result Owner	KTH
Sector(s) of application	Any systems, service and applications that need a robust secure common distributed file system with a standard POSIX file API and (optionally) policy-based security access control; e.g. grids and grid applications, content sharing, collaborative applications
Direct Stake holder	End-users: VOFS can be used by any end-users who need a common file system, e.g. for file sharing, common working space for collaboration. Developers and service providers: VOFS can also be used by developers and service providers who need a secure distributed file system to be used as a part of a distributed system or application.
Brief description	The WebDAV-VOFS (Virtual Organization File System) is a VO-aware file system based on the WebDAV protocol. WebDAV VOFS, shortly VOFS, allows VO members (users) to build a distributed file system on multiple computers donated by the VO members or external resource providers. In order to do that, VOFS allows users to expose their files and directories to a VO file system, i.e. to make them accessible (read, written, renamed, deleted and so on) within the VO. It also allows users to mount the VO file system to their local file systems, so that files in VOFS can be accessed by ordinary applications such as text editors, PDF and picture viewers, etc. In technical terms, files in VOFS can be accessed using the standard POSIX file API; it supports the complete set of standard file operations defined in the POSIX standard. VOFS includes a policy-based security, which ensures that only VO members can access files in VOFS. Access rights in VOFS confirm to VO policies set by resource (file and storage) owners. The WebDAV-based VOFS prototype with security can be used with multiple Operating Systems, such as Linux and MS Windows.
Stage of development	Laboratory prototype

Result Name	**WebDAV Virtual Organization File System, WebDAV VOFS**
Collaboration sought or offered	Use of WebDav-VOFS to develop collaborative services and applications.
Collaborator details	Service and application developers.
IPR granted or published	The 3-clause BSD license
Contact details	Leif Lidbäck <leifl@kth.se>

Result Name	**Security Infrastructure (security components, API, and tools)**
Result Owner	KTH
Sector(s) of application	Grids and Cloud platforms
Direct Stake holder	End users authenticating to a system, e.g. a Virtual Organization, protected by the security infrastructure; owners of resources; Administrators managing a Membership Service and a policy repository; Developers of software which needs security infrastructure (authentication and authorization) for access control.
Brief description	The policy-based Security Infrastructure can be used to protect distributed resources and services from unauthorized access by controlling access through authorization policies. For example, Virtual Organization (V0) members (resource owners) set VO-wide security policies, dictating how users may access VO resources. The VO security infrastructure protects resources, enforcing the VO policy. Authentication establishes the identity of a user; while authorization checks whether the use has rights to access the requested resource (e.g. a file). Authorization guarantees that the user can only access resources she has the right to access, according to policies. Authorization is policy-based; policies are expressed in XACML (eXtensible Access Control Markup Language). The security components include Policy Enforcement Points (PEP), Policy Decision Points (PDP), Policy Administration Points (PAP), Policy Information Points (PIP), and VO Membership Service (VOMS). The security infrastructure has been developed using Sun's XACML implementation (API and the policy engine). The policy-based security infrastructure is general enough to be used in any distributed system, service and application, for authentication and authorization. The policy-based authorization provided in the security infrastructure can be combined with any authentication method by replacing the default token-based method.
Stage of development	Laboratory prototype
Collaboration sought or offered	Research and test-bed Grid deployment programs.
Collaborator details	N/A
IPR granted or published	The 3-clause BSD license
Contact details	Leif Lidbäck <leifl@kth.se>

Result Name	**SIS**
Result Owner	UPRC
Sector(s) of application	Service oriented architectures, grid computing
Direct Stake holder	End-user, developer, service/resource provider , platform operators
Brief description	SIS provides matching and ranking service for peers offering or requesting resources. SIS exploits offers and requests of resources and services, in conjunction to resources' characteristics and market-related properties to satisfy peers' intentions and preferences. Resource retrieval in this context extends the notion of resource matchmaking, to the process of discovering those markets (trading instances), each with its own auction mechanism, which trade resources matching the requests of buyers and the offers of sellers. To support end-users, service developers-providers, platform operators to the discovery advertisement of services in a seamless way (i.e. without requiring knowledge of semantic annotation and matchmaking processes), adequately and effectively, a facility for the automatic translation of WSDL specifications to OWL-S profiles is provided.
Stage of development	Laboratory prototype
Collaboration sought or offered	Information exchange, organizations requiring Grid and Cloud platforms
Collaborator details	Research organizations and academia
IPR granted or published	LGPL
Contact details	George Vouros (georgev@aegean.gr)

Result Name	**P2P-LTR**
Result Owner	INRIA
Sector(s) of application	Distributed applications including collaboration tools (such as wikis).
Direct Stake holder	Developers
Brief description	Provides logging of user actions (corresponding to ACF's multilog) in a DHT and continuous, distributed time stamping of these actions. The objective of time-stamping is to perform reconciliation of replicated data in a P2P system. P2P-LTR is intended for usage beyond the project in the context of Xwiki, a client-server OSS wiki product (www.xwiki.com) developed by the French start-up Xpernet) which is used intensively for collaboration among small enterprises or within large organizations. The P2P-LTR can be used programmatically as a service and will be available as OSS to be used outside the project.
Stage of development	Laboratory prototype
Collaboration sought or offered	N/A
Collaborator details	N/A
IPR granted /publish.	LGPL
Contact details	Patrick Valduriez (Patrick.Valduriez@inria.fr)

Result Name	**eMeeting**
Result Owner	Antares
Sector(s) of application	Synchronous collaborative tool for conferencing, seminars and meetings.
Direct Stake holder	End users from medium and large-scale organizations
Brief description	eMeeting is an online synchronous collaborative tool that allows sharing not only voice and video output but also documents and charting and polling graphics. It has been extended through two service integrations, the Shared Calendar, allowing participants to fix the date of meetings and VOFS allowing participants access to a decentralized collaborative work space.
Stage of development	Prototype
Collaboration sought or offered	End-users
Collaborator details	Non-profit organizations, schools
IPR granted or published	Licence
Contact details	sergio@antares.es

Result Name	**Market place tools**
Result Owner	UPC, FT
Sector(s) of application	Market place to buy and sell computational capacity on-line. Platform for brokering, mediating between Cloud providers.
Direct Stake holder	Developers, service providers and infrastructure operators
Brief description	Tools for open market places including an open framework to develop new auction mechanisms and a decentralized information service. Two mechanisms to allocate time-differentiated resources (a) combinatorial auction for mandatory allocation patterns and (b) extended double auction for elastic applications.
Stage of development	Early prototype
Collaboration sought or offered	N/A
Collaborator details	N/A
IPR granted or published	BSD
Contact details	rbrunner@ac.upc.edu, ruby.krishnaswamy@orange-ftgroup.com

Result Name	**CFS**
Result Owner	UPC
Sector(s) of application	Computer aided learning tools
Direct Stake holder	End users in educational sector
Brief description	The application is an extension of the Firefox browser that any end user can install. This extension adds Firefox the capacity to become an interface for CFS and thus for the Grid4All environment. It contains the CFS graphical user interface, and interfaces to Grid4All data services: distributed virtual storage (VOFS) and a library (Telex) that enables remote sharing of documents and remote collaboration with semantic reconciliation.
Stage of development	Laboratory prototype
Collaboration sought or offered	Groupware researchers and developers willing to evaluate CFS in computer-supported collaborative learning environments, to improve and extend the application accordingly.
Collaborator details	Groupware researchers and software developers from universities and research-oriented organizations.
IPR granted or published	GPL Mozilla
Contact details	Joan-Manuel Marques <marques@ac.upc.edu>

Result Name	**CNSE**
Result Owner	UPC
Sector(s) of application	Computer aided learning tools
Direct Stake holder	End users in educational sector
Brief description	The CNSE is an application to carry out collaborative network simulations for learning purposes. In this sense, the current prototype of CNSE adds the following features, compared to other network simulators: Collaborative Visualization, Group Awareness and Group Management.
Stage of development	Laboratory prototype
Collaboration sought or offered	Deployment of CNSE for teaching purposes, evaluation of the application, and further research and software development on ways to work on dynamic environments.
Collaborator details	Universities and research-oriented organizations experimenting usage of e-learning tools.
IPR granted or published	GPL, Mozilla
Contact details	Carlos Alario <calahoy@gmail.com>