

# Towards an Open Grid Marketplace Framework for Resources Trade<sup>\*</sup>

Nejla Amara-Hachmi<sup>1</sup>, Xavier Vilajosana<sup>2</sup>, Ruby Krishnaswamy<sup>1</sup>,  
Leandro Navarro<sup>3</sup>, and Joan Manuel Marques<sup>2</sup>

<sup>1</sup> France Telecom R&D

<sup>2</sup> Universitat Oberta de Catalunya

<sup>3</sup> Universitat Politècnica de Catalunya

{nejla.amarahachmi,ruby.krishnaswamy}@orange-ftgroup.com,  
{xvilajosana,jmarquesp}@uoc.edu, leandro@ac.upc.edu

**Abstract.** A challenge of Grid computing is to provide automated support for the creation and exploitation of virtual organisations (VOs), involving individuals and different autonomous organizations, to which resources are pooled from potentially diverse origins. In the context of the presented work, virtual organizations trade grid resources and services according to economic models in electronic marketplaces. Thus in this paper we propose GRIMP (Grid Marketplace), a generic framework that provides services to support spontaneous creation of grid resources markets on demand. We motivate the need for such framework, present our design approach as well as the implementation and execution models.

## 1 Introduction

A main challenge of Grid computing is the creation of reliable and scalable virtual organisations on demand in a dynamic and open environment. VOs are formed of autonomous entities that are created to deliver a set of services. The formation and maintenance of VOs within an open environment is still a difficult task. In this paper we address one aspect of maintenance; on-demand resource capacity expansion as a means to adapt to fluctuating needs for computational resources in the life-time of a virtual organisation. Market based models are increasingly being studied to address resource allocation. Our objective is to provide tools and services to operate open Grid resource market places. A market based approach has two benefits: provide incentives to resource owners to share their resources and secondly provide efficient arbitration in conditions of fluctuations in supply and demand.

The contribution of this paper is to propose an architecture for GRID resources marketplace (GRIMP) that supports an environment characterised by heterogeneity and diversity of resources, applications, application behaviours, dynamicity, and scale.

---

<sup>\*</sup> Work supported by MCYT-TSI2005-08225-C07-05 and Grid4All(IST-2006-034567).

## 2 Context

This section presents a representative scenario which illustrates the requirements that drive the GRIMP framework. The scenario shows how the envisioned framework is used by VOs to adapt to changes such as that of fluctuating resource needs.

### 2.1 Scenario and Motivation

A community of vinyl record collectors creates a VO whose objective is to preserve and share their legacy of rare records. The technical objective of the VO is to execute applications that process the vinyl records. One application (A) digitizes sound and transforms it to a computer-readable format. A second (B) adds watermarks into each audio file to preserve copyrights, while the last and popular application (C) is a real-time player that plays the records and diffuses analogical sound formats to other VO members. The VO has a large member subscription most of who contribute sporadically their resources. All the three applications require processing time and the first requires storage resources as well. While a few members contribute regularly their computational resources to the VO, the majority provide them sporadically. They instead pay a subscription fee to obtain this service. In this scenario, the focus of this paper is the allocation of resources to the applications. We assume that all resources and applications of the VO are managed and the management logic takes appropriate decisions to ensure that preset goals are met. If this decision triggers resource allocation, then the self-configuration manager adopting the role of a buyer agent negotiates at the market place to acquire resources. At a time any of the applications A, B, C may have load surges and require resources to match the required quality of service. Application (A) requires both storage and processing time since the music must be digitized and stored. The execution of the digitizer is planned and scheduled by the VO administrator and hence resources are leased in advance of time. The buyer agent decides to start a combinatorial auction for processing and storage resources. Application (B) may have unplanned load surges due to remote requests by VO members to watermark files. Allocation of resources for this application is triggered by the load monitoring logic, but members may be requested to wait. The buyer agent selects a double auction that trades in processor cycles for usage within a time range provided by the application. Application (C) is stringent in its resource requirements and cannot wait for allocation. The buyer agent will select a continuously clearing double auction trading in processing time for immediate usage. The GRIMP marketplace addresses these scenarios by providing services that allow actors to spontaneously create mini and short-lived markets on demand. This places GRIMP in a design space between a decentralized and a centralized architecture, which we believe responds better to the targeted environment.

In this scenario, two different auction mechanisms to allocate computational resources are used. Although a vast range of applications require typically one type of resource as is the case of the application B and C; many others are

elastic and tolerate varying quantities of resources. Hence applications such as B and C are satisfied by mechanisms like k-double auction (a generalization of the classic first price and 2nd price auction mechanisms) and do not need computationally expensive mechanisms such as combinatorial auctions which is required by the application A. The application A needs imperatively both computational and storage resources for correct execution. Combinatory auction mechanisms though computationally expensive are required to ensure that such applications may allocate resources without confronting the exposure problem.

## 2.2 Requirements

From the scenario we derived a set of requirement for the GRIMP framework.

**Generic Infrastructure:** Instantiation of market services on need and co-habitation of multiple instances. This implies mechanisms for initiators to instantiate and configure markets when needed.

**Support for multiple market mechanisms:** Choice of different types of market mechanisms such as, combinatorial, double, English or other iterative auctions. This implies a flexible framework and tool-kit that facilitates rapid prototyping of new allocation mechanisms.

**Open architecture:** Open Grid systems are exposed to heterogeneity and constant evolution that suggests use of semantic descriptions of resources and markets to facilitate matching and discovery.

**Standardization and Interoperability:** The use of flexible standards and interoperable interfaces to facilitate interaction with external infrastructure services is required.

## 2.3 Related Work

The last years have seen a number of approaches based on economic based resource allocation within the context of Grid computing. OCEAN [1] and CAT-NETS [2] focus on a completely decentralized system based on direct negotiations between peers. Both systems demonstrate the need to provide support for multiple market mechanisms; however the aspect of interoperability of agents in the face of multiple market negotiation protocols is not addressed. Furthermore, electronic marketplaces have been extensively studied. AuctionBot [3], provides support for multiple auction mechanisms by means of configurable policies. However their approach is neither extensible nor interoperable. Rolli et.al [4] propose to break down market processes into services in order to foster flexibility. Furthermore, their framework allows the configuration and extension of market services and provides a description auction language that facilitates the development of new market mechanisms. Besides, mechanisms flow is guided by means of an orchestration language like BP4WS. GRIMP aims to propose an open, interoperable architecture for the trading of Grid resources, open APIs and layered software architecture. Moreover, an open market place for Internet level Grids needs to address both in terms of development and run-time co-habitation of multiple market mechanisms.

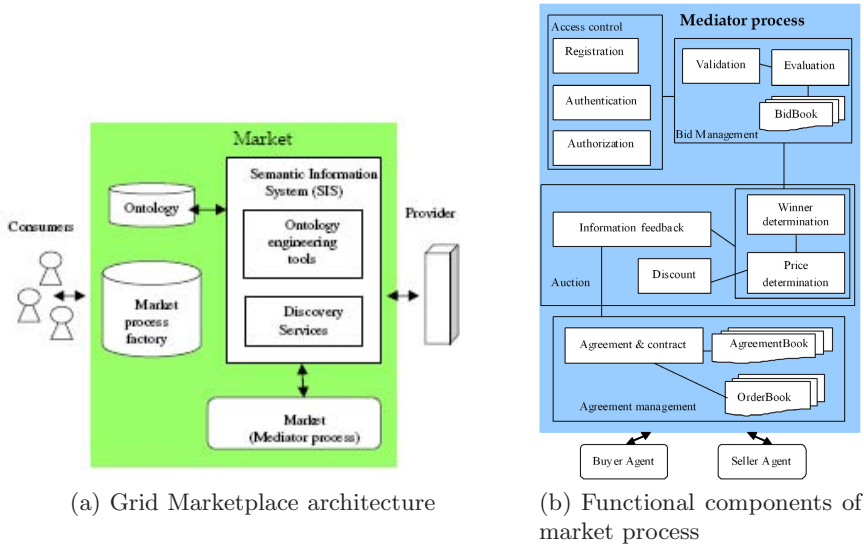
### 3 The GRIMP Architecture

This section addresses the requirements and design principals of the GRIMP framework. As part of design process, we identify common domain specific elements of an auction market so as to facilitate design and implementation of different market behaviours. The goals of the framework are reusability, extensibility, rapid prototyping of new market mechanisms. Extensibility and flexibility is addressed one the one hand by the provision of generic interfaces that allow the definition of specific mechanism following a well defined template. Customization is addressed by a protocol factory that provides functionalities to instantiate markets (given the model, type and structure based on initiator needs) and functionalities to add new market templates and new implementations (see Fig. 1a). In order to deal with complexity, the market framework is developed following a component based approach that promotes modular design, distributed and autonomous development, and reuse of the developed components. This helps saving efforts when designing new market mechanisms and higher level market applications and services. Components may be composed and assembled based on domain specific rules that constrain the composition. Besides, components encapsulate distinct aspects of the market so they can be customized and replaced independently.

One of the problems of a component based approach concerns interoperability with external services since trading sessions not only rely on the market mechanism itself but they need to interoperate with other infra-structure services such as information services for dissemination of market situations, discovery services, payments, agreement, etc. For that reason, the market process is encapsulated following a service-oriented approach. The advantage of combining both approaches is manifold, particularly as web services provide the means for software components to communicate with each other on the web using platform and language independent means, components provide a suitable approach to cope with market complexity. Buyers and sellers, need to discover such market services so as to participate within and conclude trades. To publish and discover traded Grid resources, market participants need a formal and semantic description of resources. This description should ensure a common understanding among peers and provide services to select and match (to discover markets, resource availability, etc.). [5] describes the ontology that has been developed for this purpose. This ontology provides information concerning a) the types and characteristics of the resources, b) the properties related to the specific offers and requests via which the specific resources are being traded, c) the specific properties of the markets to which orders are placed.

#### 3.1 Zoom on the Mediator Process Architecture

In this section we focus on the component-based architecture of the mediator process that we will henceforth simply refer to as market. It represents a central part in the GRIMP framework (see Fig 1b) that provides common and specific elements of the market process (economic and system tasks) allowing to implement different market behaviors.



**Fig. 1.** Grimp framework architecture and functional components

**The Market Process**

We define the Market process as the mediating process that implements a market mechanism, and based on the received bids from its participants determines the final outcome of trade between the buyers and sellers. It is created by some Trading agent wishing to buy or sell some Grid resource. The trading is guided by the rules of the negotiation mechanism encapsulated by an Auction composed of one or more rounds. During each round, the objective bids are received and stored in a local structure; at the end of the round the current set of winning allocations and the current prices of the resources are determined. Subsequently, the prices to be paid are determined after applying a Discount policy. At the end of the market process, an Agreement object is created: these associates matching pairs of bids (from seller and buyer) that have won.

**Functional Architecture**

In the mediator process architecture, there principally three types of components: market specific, system specific, and finally business platform specific. System specific components covers aspects such as registration, communication, business specific components cover aspects such as establishment of agreements, and the market specific components cover trading rules and algorithms. The Access Control component identifies and authorizes participants to register at the market. This configurable component allows the market initiator to select control policies such as limiting maximum number of traders. The Bid Management composite is a sub-component of Market and encapsulates rules governing the bidding activity. Incoming bids are validated for conformance and either stored

in waiting for clearance or dispatched to the clearing component. This composite offers interfaces that allow pre-processing of incoming bids to match the specific trading conditions of the market.

The Auction composite encapsulates the three main components: Clearing that is triggered by the Auction activity controller and matches the bids and offers that it retrieves from the Bid Manager, PricingPolicy that calculates final prices that will be paid by winners, and InformationFeedback that generates feedback quotes. The separation of clearing and discount policy permits flexibility in selection of pricing policies. This may be deferred even until deployment through adequate selection of the component contents through programmatic control. The feedback component may be independently configured by initiators of market processes to set the auction and system specific policies that govern visibility. Once the Auction component has determined the matched allocations, the Agreement component is invoked by the Market activity controller. Its role is to dispatch generated agreement records to the Agreement Manager such that contracts may be established between the matched of buyers and sellers. The separation of interfaces to the Agreement Manager allows for flexible deployment of the market mediator. The market designer (or also initiator) may select an appropriate agreement manager and establish the component bindings at run-time by using the flexible binding semantics provided by the component model.

Three additional components have been specified to store different data handled in the market such as bids and agreements. The BidBook component provides interfaces to store and retrieve bids and offers. The AgreementBook component provides interfaces to store and retrieve matching allocations decided by the Auction composite. These components may even be shared (the AgreementBook may be shared by the Clearing component and the Agreement component) between multiple components as shared state to enhance performance in particular when all sharing components are co-located on the same physical node.

## Composition, Deployment and Execution

Components can be bound following a straightforward approach consisting of a static specification through the ADL (Architecture Description Language) that describes the system composition and binding of sub-components. ADL enables also to decouple functional program development from the tasks needed to deploy, run and control the components. With this assumption, assembly may be looked upon as an off-line issue through a static fixed ADL. Nevertheless, this fixed binding of the market components is not flexible and may not be adapted to different market types and mechanisms. Two immediate alternatives can be either providing only interfaces of the framework components so that the designer has to handle by himself composition problems; or provide a meta-composition language in addition to the corresponding patterns and tools for searching and obtaining components from the components repository.

After the GRIMP framework is composed, its deployment is managed at run-time over an underlying infrastructure. Deployment can be managed by a dedicated service within or by a management layer of overlay services like the one

proposed in [6] that offers self-configuration, self-healing, self-tuning and self-protection functionalities. The Fractal API provides tools to instantiate components from ADL specifications allowing such VO service to manage and deploy components at runtime.

Deployment for a specific market mechanism is held by the **initialization** phase; this phase includes the configuration and the start-up of components. The **activity period** starts when the Market is configured and ready to accept events (registration of participants, information queries, submission of bids). This information exchange between the market and the negotiating entities is held through the external interfaces exposed by the market framework as Web Services. When the auction/negotiation has been determined to terminate according to the auction/negotiation rules, the **termination** phase prepares the agreements, ensures house-keeping activities and terminates the market.

## A Specific Implementation

In order to test the suitability of the GRIMP framework, a k-Double Auction (k-DA) mechanism has been developed. The k-DA mechanism implements the generic interfaces provided by the mediator process such as the Auction component specific interfaces and provides new functionalities to the BidManagement component. Once implemented, the K-DA lifecycle executes as follows:

**Initialization:** The GRIMP's Market factory is used to start the market. The initiator configures the market to implement a double auction with a k-pricing policy (k-DA.) that initially trades in some quantity of one particular item for a period of time. Finally its creation is advertised at the SIS.

**Activity period:** Once registered to the SIS the k-DA is prepared to receive bids. Bidders first, consult a Market Information Service (MIS) to get dynamic information about market, e.g current prices, etc... Authorization is required by the Access Control component of the Mediator process before allowing a bid submission. The Bid Management component validates and stores bids until the termination of the auction. The Bid Management component preprocesses bids to fulfil any required format. At clearing time, the Auction component executes the DetermineWinner operation of the k-DAWinnerDetermination component that computes the winning bids. The k-DAPricingPolicy and VolumeDiscount are applied to compute final prices. Once the set of winners is known, the Agreement Management component notifies the agreement to winning buyers and sellers.

**Termination:** The k-DA is terminated after the agreement is notified to both sellers and buyers. This action is also announced to the MIS.

## 4 Conclusions and Future Work

he paper proposes an architecture for a Grid resource market place that focuses on support for multiple auction formats by proposing a framework where

market rules, algorithms, and activities are encapsulated as components. We have started prototyping of this framework using the Fractal [7] model. Fractal provides the means to assemble complex markets from a set of configurable components. Several useful Fractal controllers are used in the design: The Attribute controller is used for configuration of the market, the Life-cycle controller to hierarchically start/stop components, and the Content controller to add/remove content to the sub-components. At the actual state of work, we are implementing two auction mechanisms: a combinatory auction model and a k-DA based auction both of them adapted to leasing of Grid resources. Our first objective is to maximize the reuse of components and limit the coding of market mechanism specific algorithms.

In parallel we recognize the need to focus on methodologies and tools for the design of rules of interaction. The market place consists of actors assuming different roles such as traders of Grid resources (buyers, sellers, and 3rd party mediators), auction services, agreement managers, and payment services, each of which executes a given role in the negotiation and that invoke each other through established interfaces. The interactions or conversations between the different roles are themselves guided and confined by the rules of a given negotiation protocol (such as the K-DA auction protocol).

In an open world, where functional and semantic heterogeneities exist, it is not realistic to assume that all actors converse with only one or a limited set of protocols; however being developed independently there is neither a reason to assume that any two actors of complementary roles speak the same protocol. This guides us to focus support from two points of view, firstly that of trading participants, and secondly that of designers of protocols and mechanisms. The GRIMP platform requires a Market factory of protocols that allows (a) participants (buyers/sellers/mediators) to retrieve protocol skeletons for a selected market mechanism or even verify if their own protocol is compatible with a selected mechanism and (b) developers to rapidly prototype new market protocols through protocol modelling tools that facilitate the design process and allows the designer to focus on the rules of the market mechanism.

The technological approach that fits our requirements is that of service oriented architecture. Each role in the market place enacts a business process linking components (or Web Services). BPEL processes are also Web Services and their interfaces are described using WSDL (footnote: in fact we need to consider OWL-S to be able to enhance semantically the description of capabilities). For example, in the case of the auction service the WSDL specifies the operations that may be invoked (such as SubmitBid, Register, Query, etc.). We are evaluating WS-CDL to specify top-down the choreography that will serve to generate role behaviour descriptions as BPEL processes. A technical issue that we face in the architecture is that of bridging the gap between the component based market framework and that of execution of the service as a business process.

Our plans for future work are driven in the following main directions: (a) achieve the design of several auctions and market protocol formats within the framework to validate the architecture (b) provide tools to assemble components



implementing a specific set of market rules (c) address compatibility checking and adaptation of participant behaviours to active market instances.

## References

1. Padala, P., Harrison, C., Pelfort, N., Jansen, E., Frank, M., Chokkareddy, C.: Ocean: The open computation exchange and arbitration network, a market approach to meta computing (2003)
2. Eymann, T., Reinicke, M., Ardaiz, O., Artigas, P., Freitag, F., Navarro, L.: Decentralized resource allocation in application layer networks. *ccgrid* 00, 645 (2003)
3. Wurman, P.R., Wellman, M.P., Walsh, W.E.: The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In: *Proceedings of the 2nd International Conference on Autonomous Agents* (1998)
4. Rolli, D., Luckner, S., Momm, C., Weinhardt, C.: A framework for composing electronic marketplaces from market structure to service implementation. In: *WeB 2004. Proceedings of the 3rd Workshop on eBusiness, Washington D.C.* (2004)
5. Kotis, K., Vouros, G., Valarakos, A., Papasalouros, A., Vilajosana, X., Krishnaswamy, R., Amara-Hachmi, N.: The grid4all ontology for the retrieval of traded resources in a market-oriented environment. (submitted to: *Service Matchmaking and Resource Retrieval in the Semantic Web Workshop Korea*) (November 2007)
6. Brand, P., Hoglund, J., Popov, K., de Palma, N., Boyer, F., Parlvanzas, N., Vlassov, V., Al-Shishtawy, A.: The role of overlay services in a self-managing framework for dynamic virtual organizations. In: *CoreGRID Workshop on Grid Programming Model Grid, P2P Systems Architecture, Greece* (June 12-13, 2007)
7. <http://fractal.objectweb.org/>