BEinGRID: Development of Business Models for the Grid Industry

Katarina Stanoevska-Slabeva[1], Davide Maria Parrilli[2], and George Thanos[3]

[1] Mcm Institute of the University of St. Gallen, Blumenbergplatz 9, CH-9000 St. Gallen
[2] Interdisciplinary Centre for Law and ICT, K.U.Leuven – ICRI -IBBT,
Sint-Michielsstraat 6, B-3000 Leuven
[3] Network Economics and Services Group, Athens University of Business and Economics,
76 Patission Str. Athens, Greece
Katarina.Stanoevska@unisg.ch,
davide.parrilli@law.kuleuven.be, gthanos@aueb.gr

Abstract. Driven by the increasing demand, grid technology is entering the business market in form of utility computing, grid middleware and grid-enabled application. However, the business market is interested in complete grid solutions. This means that for a successful take up of grid technology on the business market the establishment of grid value networks is required. This again can only be achieved by implementation of sound business models for each player providing part of a grid solution. This paper discusses the business models of providers of grid-enabled application.

Keywords: Grid Business Models, Business Grids, Grid enabled Application.

1 Introduction

Newest market research studies report a growing awareness for the potential of grid technology by industry and increased interest for utility computing and grid solutions for business application. This trend has been in particular enforced by well established Internet companies as for example WebEx, Amazon, AOL, who offer their services in form of utility computing [1]. Another player driving utility computing are telecommunication companies. For example T-Systems in Germany is rolling out in cooperation with SAP an SAP on-demand service. A growing interest for grid computing can also be observed with Independent Software Vendors (ISV) [1]. This is mostly evident in vertical markets with strong grid interest or for applications that are suitable for grid (for example data mining). First steps towards grid friendly licensing models can be observed at some vendors even though there is the fear of cannibalizing existing business models for packaged application.

Driven by the growing interest and demand on the market, grid technology is entering a new level of maturity and is offered on the business market in three forms [3]: 1) as open source or packaged grid middleware; 2) as utility computing, that is as hardware and software infrastructure provided according to the Software as a Service (SaaS) paradigm, and 3) in the form of grid enabled application. However, business

J. Altmann, D. Neumann, and T. Fahringer (Eds.): GECON 2008, LNCS 5206, pp. 140–151, 2008.

customers are interested in complete grid solutions. This means that for a successful take up of grid technology on the business market the establishment of grid value networks [15] is required that will be able to provide complete solutions and a critical mass of offerings on all levels of the value network. This again can only be achieved by implementation of sound business models for each player providing part of a grid solution. In particular, new business models are required from two perspectives: the grid utility computing providers and providers of grid enabled applications. The business models of these two players of the grid market are closely related to each other. On the one hand grid enabled application are an important driver for the demand of grid resources offered as a service. On the other hand grid infrastructure offered as a service is a necessary prerequisite for grid enabled application. Thus, a critical mass of grid enabled application is needed for the next step of the grid market evolution. However, while there is a growing body of literature on business models or specific components of them for the utility computing market [3], [15], there is less consideration of business models from the perspective of providers of grid enabled applications. This paper provides a contribution in this context and discusses the main aspects of business models of ISV evolving their products from pre-packaged applications towards grid-enabled application.

The content of the paper is structured as follows: Section 2 provides an overview of definitions and the research approach. Section 3 provides an overview of business models of pilot applications developed as part of the BEinGRID project. Section 4 provides a generic concept for components of business models for grid enabled application. Section 5 concludes the paper with a summary and outlook.

2 Research Approach

The research presented in this paper followed the following approach:

1. First the most important terms (grid-enabled applications and business models) involved in the research were defined and an analysis approach was chosen.
2. Then business models of technology and application providers were analyzed based on case studies of grid pilots from the project BEinGRID. BEinGRID (www.beingrid. com) is an Integrated Project (IP) that is funded by the European commission under FP6. One of the main objectives of the project is to evaluate the applicability of grid technology in business through grid business experiments. In the heart of the project there are 18 business experiments that are piloting grid technology in various key industrial sectors. In this paper the business models of pilots focusing on grid-enabling application were analyzed.
3. Finally the findings of the analysis were aggregated to a generic business model for providers of grid-enabled application.

The resulting business model can be applied by providers of grid-enabled applications as a checklist for developing successful business models.

2.1 Grid-Enabled Application Definition and State-of-the-Art

The term grid-enabled application is used in this research paper to denote software application that have been offered on the market as pre-packaged software and that are being extended in a way that they can run in a distributed manner in a grid environment. To grid-enable a pre-packaged software product therefore means that a previously pre-packaged centralized application is enabled to run either on a distributed grid infrastructure or to be offered as an online service based on the Software as a Service paradigm (SaaS) (see also [4]).

In principle, the idea of providing applications in a SaaS manner is not a new concept. A similar concept for software delivery was introduced by [5] in 1998 under the term "Application Service Provisioning (ASP)". ASP evolved from IT outsourcing and is based on the idea that a web-enabled application can be provided online through IP-based telecom infrastructure [6] by a central application service provider [9]. At the beginning the ASP model was a typical one-to-many delivery model, which means that the application is operated in a centralized manner by the application service provider and is offered in the same manner to many customers. The main advantages of the ASP business model for customers are: cost savings and no need for developing and maintaining an own infrastructure and skills.

Even though at the first glance the business models of ISVs offering grid-enabled application and of ASPs seem similar, there is a significant difference. The core competence of the ISP is the development of the application itself and not its distribution. On the contrary, the core competence of the ASP is the online provision of applications that are mostly developed by other ISV. Despite of the difference regarding their business models, key learning's from the experiences with the ASP business model can be applied during development of business model for grid-enabled application. Even though ASP was foreseen to be successful, it did not take up on the market and its adoption has been very slow [7]. The main reasons for the failure have been: the inability of early ASPs to produce customized services, the centralized approach for computing, which requires the sending of input and output data and the general lack of trust in the ASP paradigm [6], [7], [9].

At present, the business models of grid-enabled application and ASPs are converging. The convergence of web services and grid computing technologies is expected to solve current ASP delivery problems [6], [9]. The ASP business model is evolving from one-to many to a many-to-many model, where several service offerings are bundled and can flexibly be applied by the user [7].

2.2 Definition of Business Models and the Business Model Analysis Framework

There is a considerable body of literature related to business models. The definitions of business models range from very broad ones as for example the definitions proposed by [10] or [11] to very specific ones (see for example [12] or [13]). [10] for instance defines in a most basic sense business models "*as the method of doing business by which a company can sustain itself - that is, generate revenue*". While such definitions try to delimit the scope of the meaning of the concept business models, they do not provide insights into components of business models in such a way that it can be used for assessing the activities of a company in more detail. A more concrete definition is the definition of Timmers [14]. According to Timmers, a business model is "*… an architecture for the products, services and information flows, including a description of various business actors and their roles, a description of the potential benefits for the various business actor, and a description of the sources of revenues."* [14]. The definition provided by Timmers was used as starting point for the development of the so called MCM business model analysis framework. The MCM-Business Model Framework provides a generic overview of components of business models based on Timmers that need to be considered during a business model analysis or design. It has been used successfully for structuring the analysis of business models of different type of digital products [15]. The components of business models denoted by Timmer's definition were extracted and enhanced with further aspects affecting business models (for example "Social Environment"). Further components of business models have been synthesized based on an in-depth analysis of the body of literature about business models [10], [11], [12], [14]. The resulting MCM-Business model analysis framework is presented in figure 1:

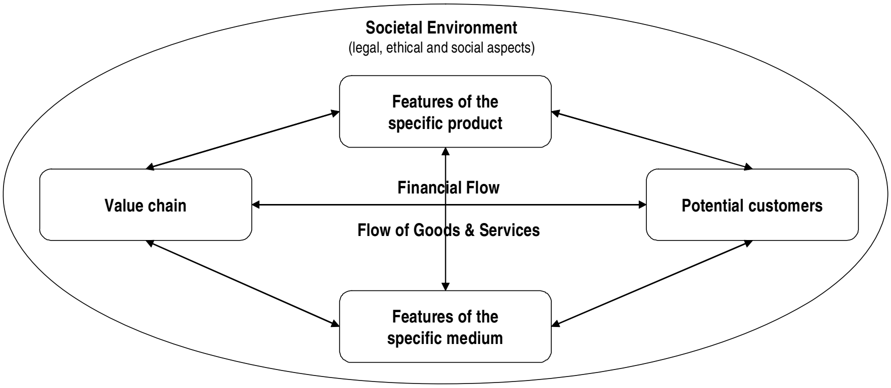
Societal Environment
(legal, ethical and social aspects)
Features of the specific product
Value chain
Financial Flow
Flow of Goods & Services
Potential customers
Features of the specific medium

Fig. 1. MCM-Business Model Analysis Framework

The elements of the framework that need to be considered during analysis or design of a business model are explained in more detail below:

The **social environment** component of a business model reflects all outside influences on the business models, such as the legal and ethical aspects as well as the competitive situation in the market. It refers to the social and regulatory context in which a business model is developed and implemented.

The component **features of the medium** expresses the possibilities for transaction and interaction over a specific medium. For example different applications are possible based on grid or a centralized infrastructure.

The component **potential customer** covers all aspects of target groups and customers as well as the expected added value provided by the product or service subject of business model development. The different business models certainly address different target groups, and do address different needs of the customer. Choosing the right target customers and designing the product according to their needs are key success factors.

The component **value chain** reflects the involved players necessary for the production and delivery of the offered product or service and their interrelationships. A typical grid value network consists for example of a content owner, content aggregator, content provider, portal owner and of course the user (for a complete generic grid value chain see [15]).

The component specific **features of the product** express the exact design and the way the service is experienced by its customers. It also explains what the specific benefits are, and how the customer might be contributing.

The component **financial flow** explains the earning logic of the business model and makes it clear which elements of the value chain contribute from a financial perspective.

The component **flow of goods and services** identifies all the processes within the company and the value chain necessary for the creation of the product or service.

The components of a business model are interrelated among each other. For example the target group of customers and their needs is influencing the product design. The product design requires a certain value network and also needs to consider legal and ethical requirements. The agreed upon relationships among the involved players of the value network are the foundation for the financial flow and the flow of goods and services. The different components need to be smoothly integrated into a business model that offers the opportunity for sustainable business and profit for all involved players.

3 Case Studies of Grid-Enabled Applications Business Models in the BEinGRID Project

The business model analysis framework was applied for an in-depth analysis of the intended business models of ISVs participating in experiments of the BEinGRID project and developing a grid-enabled version of their product. Out of the 18 pilots six are aiming towards business models for grid-enabled application:

- Business experiment (BE) BE16 has developed a grid-enabled extension of an existing application for ship design and simulation so that it can be offered in cooperation with an infrastructure provider in a SaaS manner.
- BE18 grid-enabled an existing application for processing of seismic data and plans to offer the service over the Internet in particular to small and medium size enterprises.
- BE07 grid enabled an existing application for generation of global aerosol maps using information coming from different satellite sensors.
- BE03 has grid-enabled an application for 3D rendering and animation.
- BE12 and BE17 are grid-enabling existing application for supply chain management.

The in-depth analysis of the business models of the above BEs has revealed several advantages and obstacles that need to be considered during the design of the business model. The main advantages are: From the perspective of the ISV the enhancement of existing application clearly provides a valuable extension of the existing application portfolio. In addition to that most of the above BE can achieve a broad competitive advantage, as most of them can leverage a first-mover advantage. In particular for the small ISV (BE12 and BE17) to grid-enable their application provides a clear competitive advantage and also a needed precondition to stay on the market. To offer the grid-enabled version of the application also results in an image gain for the companies. For most of the companies the grid-enabled version of the application is applied to approach a new category of target customers - small and medium size companies.

The main obstacles that need to be overcome are the following: At present all providers of grid-enabled application need to establish sound business relationships with utility computing providers, in order to be able to offer a complete solution. This means that the establishment of the whole value chain is necessary. Another major obstacle is the fear of cannibalization effects for the existing centralized application. As the described application show, the applications that are being grid-enabled are applications that are needed by the customer companies occasionally. This means that by talking advantage of a SaaS offering customers might try to optimize the usage and pay less than for the licenses for the centralized application.

The above findings have been considered for the development of generic business models.

4 Development of Generic Business Models for Grid-Enabled Application

Based on the findings from the case studies general guidelines for the development of the business models were developed. Considering the above obstacles the main emphasize in this paper was on the following components of the business model: design of the product, design of the value chain and legal aspects. The findings are explained in more detail in the sections below.

4.1 Design of the Product

The design of the grid-enabled application needs to address in particular the cannibalization problem. A careful strategy is necessary, in order to keep existing customers that do not want or cannot use the grid-enabled application and to meet the requirements of new customers (see also [16]). An important question is: Are different versions for different customer segments and licensing strategies possible and in which form? The problem can be illustrated on the following example:

> **Example:** One ISV offers an application with a given set of functions to the market. A grid-enabled version of the application is developed. However not all of the existing customers have a grid infrastructure and cannot apply the new functionality. They would like to stay with the centralized version of the application. A small number of the customers has already an own grid infrastructure and would like to take advantage of the new functionality. This are also the customers that have a high volume of transaction and would also be willing to pay more for the enhanced application. The ISV gets furthermore requests by smaller companies for an occasional use of the application based on the SaaS paradigm. After a certain time a cooperation with a grid infrastructure provider is agreed upon and the application is also available on a SaaS basis.

The question now is how the different categories of the products should be defined and which licensing and pricing strategy should be defined? A low price for the SaaS application might result in the effect that existing customers of the centralized application - in particular those that use the application occasionally - switch to the SaaS application and save the licensing costs for the central version of the application. In order to avoid such effects, a carefully designed packaging of the functionality of the different versions of the application together with the licensing and pricing strategy is necessary. The different options regarding versioning of the products are discussed below.

Versioning option 1: Offering the application in form of commercial software with and without grid enhancement and without SaaS option (c.f. 2):

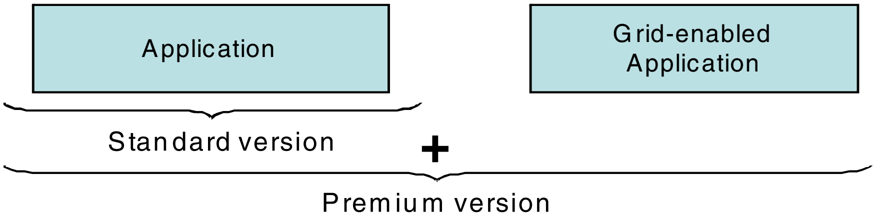

Fig. 2. Standard and premium version of a grid-enabled application sold as commercial product

The versioning example given in figure 2, enables to keep the existing customer base and the established licensing models for the existing application and provide a premium version for customers that have an own grid infrastructure. This versioning option provides the basis for diversified licensing strategies, to target customers with different needs as well as for additional revenues as the grid-enabled application can be offered with adding additional licenses for it.

Versioning option 2: In case the application is available as centralized application, grid-enabled application and SaaS, several different options for versioning and packaging are possible. One possible example is given in figure 3 below.

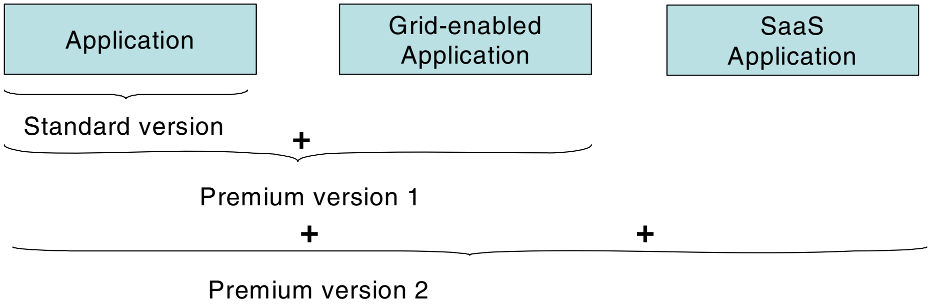

Fig. 3. Example of versioning strategies based on three product categories

As in versioning option 1 there might be a standard and a premium version if it is bought by the customer together with the grid enhancement. The question here is how the SaaS version can be included in a way that it might be suitable to also attract new customers, for example SMEs that cannot afford the premium 1 version, but at the same time not provoke a massive switch from the lucrative licenses for the central application by existing customers. One option is that customers opting for the premium version 1 can add also access to the SaaS version and pay additionally per use, if in addition to their own grid they use also the SaaS. A similar option might be available for the customers with the standard version of the application. The question is how to differentiate the SaaS version. One possibility would be to limit the functionality of the SaaS version or to differentiate a "light" version with respect to the output options or other functionality that are available. For example, an SME that wants the functionality as SaaS might get the output data only in a basic format, while premium customers get it in a pre-defined format. Similar differentiation of the quality of the service can be made also based on other features of the product and service (see also [16]). How exactly the existing functionality can be packaged in the three product categories and which versions are possible depends on the modularity of the software, the existing customer base and the potential for segmentation of the customers depending on their willingness to pay and their specific needs. A good knowledge of the usage patterns of customers as well as their willingness to pay is therefore a clear advantage in determining the right versioning and pricing strategy.

4.2 Price Strategies of the BEinGRID Business Experiments

The pricing strategy involves two components: the pricing model and the definition of the prices. The major general pricing models for grid enabled applications are Pay-per-use pricing models. Thereby the price might include infrastructure and access to the application or be provided separately for grid computing infrastructure and for the application. In this pricing model the price per usage includes also the license. A benchmark for pricing in this area might be the published price of SUN of 1$/hour computing resources or the pricing strategy of Amazon: 0.20$ per GB stored or to hire a complete virtual PC for $0.10 per hour.

The definition of concrete pricing depends on the specific product. In case where different versions of the product are involved, pricing should not affect the product strategies. For example: BE01 found during the competitive analysis that a license for computational fluid dynamics software can vary from £10'000 to £15'000 per single CPU license and go up to £100'000 for 64 CPUs. In case such an application is grid enabled, the question is what the right price might be. Several aspects need to be considered: The typical usage patterns of an average customer, the market prices for similar services and the costs of the provider. For example let's assume that in case of SaaS the same number of CPUs is used. How can the license per CPU be expressed per hour of usage? If a price that is too low is chosen than the ISV does not have interest to provide the application as SaaS as he will lose revenue. In case data about the usage patterns of customers are available the actual average usage per year could be transformed in a price. For example, the provider knows that an average customer is using the application 50% of a person's yearly working time per user. This would mean that the application is used by a typical user for 840 working hours (assuming a yearly total of working hours of 1680). Thus, in order to get the same revenue from the user based on a SaaS version of the application a price of £12 per hour for the application would be required (assuming a basic license of £10'000 for a single CPU). In a similar way based on average usage patterns and total number of users a potential price might be calculated.

4.3 Design of the Value Chain

As mentioned above, in order to bring a grid-enabled application on the market, it is necessary to assure the availability of a grid infrastructure by bundling it with offerings of utility computing. The ISV can achieve this in two ways - either by developing know-how and deploying an infrastructure by himself or by partnering with a provider of utility computing. Option two has obviously more advantages. However, it cannot be implemented in all cases. For example BE12 and BE17 are very small companies and have small customers and are located in Italy so that a low volume of total transaction can be expected. Such a low volume of expected transaction is not relevant for the utility computing provider, so that a partnership could not be established. The ISVs need to provide a grid infrastructure themselves.

In case a partnership can be established, an important design option is the question who of the two players will orchestrate the offering and have the customer ownership [see also 17]. The application provider should strive towards partnerships where he can keep the customer ownership.

4.4 Legal Aspects

The analysis of the project's cases shows that in addition to business aspects, major legal issues have to be addressed as well [18].

It is pivotal to address, as starting point, what is, in legal terms, the agreement that encompasses the provision of SaaS. This, of course, depends on the applicable national legal framework but, in general it means to set up an ASP contract. The provision of SaaS implies that there is no physical item delivered to the end user and that, unlike in the contract between a customer and a software house for the writing of a specific computer programme, the software provider keeps the ownership of the application. In case of due diligence, for instance, this element has to be taken into account, as the software can be considered as an asset (and not a liability) of the targeted company only if this undertaking has the ownership of the software.

The service provider will limit as much as possible the rights of the client, which could use the SaaS only during its ordinary course of business, thus he will be liable for breach of contract if, in practice, he sublicenses the supplier's applications. It is pivotal to say that the parties, by virtue of their contractual freedom, would have the possibility to adapt the above clause to their exigencies, and they could opt, for instance, for a transferable or exclusive license. As regards the code provided to the client, in a typical SaaS scenario the object of the contract will concern the object code and not the source code.

The contractual freedom of the parties plays a fundamental role also as regards confidentiality obligations. This issue is particularly complex and the experience gained shows that the relative clause should address at least the following issues:

- Extension of the confidentiality obligations of the supplier and the client as regards, basically and respectively, the data of the customer and the executable code of the software;
- Duties of the parties;
- Contractual and Court remedies, taking into account that the latter are heavily influenced by the applicable national legal framework;
- Exceptions to the rule, i.e. situations in which there are no confidentiality obligations.

We have developed the following template that encompasses the abovementioned elements and that is suitable to be adopted in case of SaaS in a Grid environment: “Customer shall not sell, transfer, publish, disclose, display or otherwise make available any portion of the executable code of the Application to others. Client agrees to secure and protect the Application and the Service in a manner consistent with the maintenance of Supplier’s rights therein and to take appropriate action by instruction or agreement with its users to satisfy its obligations hereunder. Client shall use its best efforts to assist Supplier in identifying and preventing any unauthorised access, use, copying or disclosure of the Application or the Service, or any component thereof, or any of the algorithms or logic contained therein. Without limitation of the foregoing, Client shall advise Supplier immediately in the event Client learns or has reason to believe that any person to whom Client has given access to the Service has violated or intends to violate the confidentiality of the executable code of the Application or the proprietary rights of Supplier, and Client will, at Client’s expense, cooperate with Supplier in seeking injunctive or other equitable relief in the name of Client and Supplier against any such person.

Client agrees to maintain the confidentiality of the executable code of the Application using at least as great a degree of care as Client uses to maintain the confidentiality of Client’s own confidential information (and in no event less than a reasonable degree of care). Client acknowledges that the disclosure of any aspect of the executable code of the Application, including the documentation or any other confidential information referred to herein, or any information which ought to remain confidential, will immediately give rise to continuing irreparable injury to Supplier inadequately compensable in damages at law, and Supplier is entitled to seek and obtain immediate injunctive relief against the breach or threatened breach of any of the foregoing confidentiality undertakings, in addition to any other legal remedies which may be available. In addition, Supplier may immediately terminate this Agreement, including all license rights granted herein, in the event Client breaches any of its confidentiality obligations regarding the Application or the Service.

Furthermore, Supplier agrees that it shall not disclose to any third party or use any information proprietary to Client including information concerning the Client and the users, trade secrets, methods, processes or procedures or any other confidential information of the other party which it learns during the course of its performance of the Service, except for purposes related to Supplier’s rendering of the Service to Client under this Agreement or as required by law, regulation, or order of a court or regulatory agency or other authority having jurisdiction thereover. In addition, Client may immediately terminate this Agreement in the event Supplier breaches any of its confidentiality obligations set forth herein. Notwithstanding the foregoing, the confidentiality obligations set forth in this Article will not apply to any information which the recipient party can establish to have: (i) become publicly available without breach of this Agreement; (ii) been independently developed by the recipient party outside the scope of this Agreement and without reference to the confidential information received under this Agreement; or (iii) been rightfully obtained by the recipient party from third parties which are not obligated to protect its confidentiality."

It is furthermore interesting to consider and define the liability of the software supplier. In this field, in fact, the ASP agreement (and the other related contracts entered into by the concerned parties) has the duty to shift and balance the risk and the corresponding liabilities between the software provider, the Grid provider and the end user. In principle, in fact, the former should avoid to be liable (if it does not own and manage the Grid infrastructure) for technical failures of the Grid itself. In other words, he should be liable only for deficiencies that are under his control. At the same time, provided that the majority of disputes concern the gap between the concrete performance of the service and the level expected by the client, the use of Grid technology should reduce this risk and, at the same time, as explained above, could extend the burden of liability of the software provider. For this reasons, the software provider should limit his responsibility to the functionality of the application and the service to the exclusion of the client's requirements. As regards the remedies at disposal of the customer, then, they usually include Service Credits (and, with this regard, it is possible to wonder whether the customer, in a Grid environment, will require higher credits in case of failure to meet the promised level of services), damages (regulated by the applicable national laws) up to, in the most serious cases, termination of the contract.

5 Summary and Conclusion

The goal of the paper was the discussion and development of a generic business model framework for providers of grid enabled application. Based on five in-depth case studies first major advantages and obstacles for developing business models for grid-enabled application were identified. Then following the business model analysis framework, general guidelines for the design of the product, the value chain and the legal issues related to provisioning applications in a SaaS manner have been developed. The core consideration has been the avoidance of cannibalization efforts with centralized applications.

Acknowledgement. The research presented in this paper was part of the project BE-inGRID (034702), which is supported by the European Commission in the sixth Framework Program.

References

1. The 451 Group, Grid computing preview, Section 7.2 in Review/Preview 2006-2007, The 451 Group, pp. 83–89 (2007)
2. MomentumSI, Implementing a Successful Service-Oriented Architecture (SOA) Pilot Program, Actional Corporation (2005)

3. Forge, S., Blackmann, C.: Commercial Exploitation of Grid Technologies and Services - Drivers and barriers, Business Models and Impacts of Using Free and Open Source Licensing Schemas. Final Report of the European Study No. 30-CE-065970 /00-56
4. Snjeepan, V., Matsunaga, A., Zhu, L., Lam, H., Fortes, J.A.B.: A Service-Oriented, Scalable Approach to Grid-Enabling of Legacy Scientific Application. In: Proceedings of the IEEE International Conference on Web Services (ICWS 2005) (2005)
5. Heart, T., Pliskin, N.: Is e-commerce of IT application services alive and well? Journal of Information Technology Theory and Application 3(4), 33–41
6. Xu, H., Seltsikas, P.: Evolving the ASP Business Model: Web Service Provision in the Grid Era. In: Proceedings of the Second Intzernational Conference on Peer-to-Peer Computing (2002)
7. Desai, B., Currie, W.: Application Service Providers: A Model in Evolution. In: Proceedings of the ICES 2003, Pittsburgg (2003)
8. Stanoevska-Slabeva, K., Talamanca, C.F., Thanos, G., Zsigri, C.: Development of a Generic Value Chain for the Grid Industry. In: Veit, D.J., Altmann, J. (eds.) GECON 2007. LNCS, vol. 4685, pp. 44–57. Springer, Heidelberg (2007)
9. Mittilä, T., Lehtinen, K.: Customizing the Application Service Provider Offering. (2005), http://www.ebrc.fi/kuvat/1066.pdf
10. Rappa, M.: Managing the Digital Enterprise.(2005), http://digitalenterprise.org/index.html
11. Afuah, A., Tucci, C.L.: Internet Business Models and Strategies. McGraw-Hill, New York (2001)
12. Osterwalder, A.: The Business Model Ontology. Ph.D. Thesis at the HEC Lausanne (2004)
13. Staehli, P.: Geschätsmodelle in der digitalen Ökonomie. Josef Eul Verlag, Lohmar, Köln (2002)
14. Timmers, P.: Business Models for Electronic Markets. International Journal on Electronic Markets and Business Media 8(2), 3–8 (1998)
15. Hoegg, R., Stanoevska-Slabeva, K.: Towards Guidelines for the Design of Mobile Services. In: Proceedings of the ECIS 2005 conference (June 2005)
16. Shapiro, C., Varian, H.R.: Information Rules - A Strategic Guide to the Networked Economy. Harvard Business School Press, Boston Massechusets (1999)
17. Tapscott, D., Ticoll, D., Lowy, A.: Digital Capital - Harnessing the Power of Business Webs. Harvard Business School Press (2000)
18. Raysman, R., Brown, P.: Computer Law: Drafting and Negotiating Forms and Agreements. Law Journal Press, New York (2003)