2000

Business Models for ASP Marketplaces

Gerrit Tamm
Humboldt University, gtamm@wiwi.hu-berlin.de

Oliver Guenther
Humboldt University, guenther@wiwi.hu-berlin.de

Business Models for ASP Marketplaces

Gerrit Tamm and Oliver Günther*
Institute of Information Systems
Humboldt University
Spandauer Strasse 1
10178 Berlin, Germany
{gtamm|guenther}@wiwi.hu-berlin.de

Abstract: ASP (Application Server Provider) marketplaces provide a fundamental alternative to the classical business model of software licensing. At this point, it is still unclear why and when customers prefer the ASP model over more traditional approaches. To make ASP more attractive, more knowledge about possible pricing and product strategies is needed. In this paper we describe different business models for ASP marketplaces. We first compare the cost structures of the classical licensing model with the new server-based approach. Then we illustrate how price and product differentiation may improve overall market efficiency. In particular, we show that by selling different software versions for different prices, ASP marketplaces may obtain near-optimal revenues with products that are relatively inexpensive, disaggregated, and customizable. Consumers can thus choose between a wide variety of product lines to fit their differing budgets and requirements.

1. Introduction

ASP marketplaces offer customers access to a personalized computing environment from any networked computer. Customers have ubiquitous access to all their files, applications, and to their corporate network. By offering customized online environments, small- and medium-sized companies can get wide area networking, remote access, and file servers for a price that scales to the size of the organization [17]. The cost of using software online may be reduced dramatically because the customer does not need to purchase dedicated hardware or worry about in-house expertise for installation and maintenance. Server-based computing helps to reduce the total cost of application ownership (TCO) by managing business-critical applications centrally while leveraging local computing infrastructure. The classical market structures will be expanded by new relations between the market actors. ASP marketplaces will take on a central role in this new business area. These developments are supported by plans of several major software companies, including Microsoft, Oracle, and HP, who will offer the next generation of their software packages for online use on an ASP platform [9].

Before this kind of software marketplaces will become commonplace, however, several critical questions remain open to research. The main problem is not the development of technical solutions, such as micropayment schemes or secure networks. It is not a technical but rather an economic problem that software providers have to face. They need to renew and extend their product lines and pricing schemes for online business [6]. It will not suffice to simply transfer the classical software licensing model to the online medium. New business concepts, which allow the customization and personalization of software components in a product line (*versioning*) and offer personalized desktop configurations will determine over success or failure of a software marketplace [18].
In this paper we are going to investigate relevant aspects for the success of an online marketplace for digital products. We will start by describing the marketplace, its products, and its actors. Following we will compare the cost structure of classical desktop and client/server architectures with the ASP approach. Then we will illustrate new pricing and product strategies in the context of marketplaces for digital goods. We conclude with ideas for future research, in particular concerning the readiness of customers to use or not to use an ASP or software market places.

2. Market Intermediaries and ASPs

Since 1995, we have been developing an electronic market for application service providers, called *MMM (Middleware for Method Management)* [12, 17]. It offers an infrastructure for managing the deployment and use of distributed application services on the web. Applications reside and execute on the software application provider's platforms, but are managed through the MMM infrastructure. MMM provides customers with universal access to all kinds of software applications, regardless of their client hardware, operating platforms, or network protocols.
MMM is a decision support system for software applications in that it helps users to select software

* also with Pôle Universitaire Léonard de Vinci, Paris, France

components suitable to match their individual requirements. By using MMM, organizations can keep their existing infrastructure while deploying recent software applications across the enterprise. Applications running on some remote server look, feel, and perform as though they were running locally. MMM is an efficient computing technology that separates the application's logic from the user interface, so only keystrokes, mouse clicks and screen updates travel through the network. Centralization of the client- and application management enables large computing environments to overcome the critical application deployment challenges of management, access, performance and security. Deploying enterprise applications is traditionally time-consuming, complex and expensive. It requires administrators to physically distribute applications to every client, plus manage version control issues, user support, multiple systems configurations and data replication. MMM simplifies and enhances application management by enabling deployment, administration and support from a single point. Updates and additions are made only once – at the server.

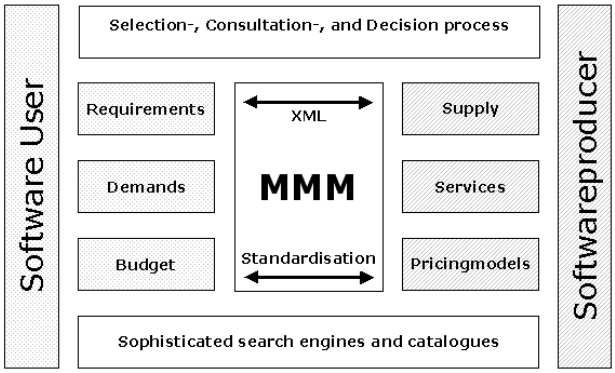

Figure 1: MMM - Middleware for Method Management

The aim of MMM is to provide a software marketplace solution for small and medium sized companies. It enables companies to use software components online or to outsource selected processes. Thereby companies can focus on their core business and reduce their costs for IT personnel, software licenses, and application maintenance. MMM propagates a paradigm that involves new pricing and product strategies like differentiated pricing for particular customers and versioning the product for individual customer requirements. The new pricing and product strategies are based on a complex business model which allows software producers or dealers to offer their software products in the classical licensing model or in more adapted models like leasing, renting, or pay-per-use [17]. MMM users can interact with the application services through a standard Internet browser, not requiring any additional software. Costs for local hardware, implementation, system maintenance, and IT personnel can be reduced significantly by using infrastructures such as MMM. To reduce the total cost of ownership, organizations have to leverage their computing infrastructure, applications, networks, and training.

2.1. Function of the online software marketplace

Intermediaries like MMM play a central role in the new digital economy and in the exchange of digital products and information. They create economic value for customers, sellers, market intermediaries, and for society at large.

Matching requirements of buyer and seller	
Software offering – software features – software lines – aggregation of software **Price discovery** – software auctions	**Search functions** – software price and information – buyer preferences – community information **added value for customers** – community information
Facilitation of trade	
Personalization – customized environment – software lines – on demand software integration **Payment systems** – SET	**Executing software Online** – no installation – no service – no hardware **Trust** – secure data transaction
Providing secure & institutional infrastructure	
Standardization – XML EDI **Legal** – contract law – intellectual property protection	**Monitoring** – data mining – regulations **Front-end access** – PDA, mobile Phone

Figure 2: Function of the online software marketplace

Intermediaries have three main functions (Fig. 2): (a) matching the requirements of buyer and seller; (b) the facilitation of trade, exchange, and payment of digital products; (c) providing a secure institutional infrastructure that reinforces the efficient function of the market [2].

The main function of an intermediary is the process of matching buyers' demand with sellers' product offerings. For this process the software seller must determine and present the product offer to buyers in a transparent way. Intermediaries can provide vendors with information about demand for special software versions or aggregations of software components [3]. By doing so, intermediaries will stimulate the suppliers to develop new product versions with characteristics that match the needs of buyers. Buyers and vendors can reduce their search costs by using an intermediary [4]. It accelerates the matching process through price discovery determining the prices at which demand and supply clear and trade occurs. More detail about the impact of the pricing strategies for an intermediary will be given in Section 4.2.

2.2. Actors and their roles in ASP marketplaces

Application Service Providers (ASPs) own the application or a special software license to offer the software product online. An example for an ASP is the company Thinter.net.

Data Providers publish and offer data and access to databases. They may simply submit files to an online repository, or provide more sophisticated online data source that ships or pushes data on demand. Examples include online stock quote services, geographic

information systems, and consumer pricing data services [17].

Application Server Providers host the computational services (application) and provide network access to them. Application server providers and application service providers will often coincide.

Infrastructure Providers or ***Intermediaries*** offer a framework that grants authorized users secure access to application server sites. They act as an intermediary between the buyer and the seller, which makes them responsible for the trade function of the online software marketplace (Section 2.1). Consequently they can be considered as the kernel of the new software market structure. The infrastructure provides functions to establish secure connections, to interoperate applications, and to manage services remotely. It may also offer new pricing models such as pay per usage and software renting in addition to the classical business model of licensing. Moreover it offers new product strategies such as versioning and disaggregation of software suites.

Computational service infrastructures, such as MMM, provide a wide range of services for different business models (licensing, renting, advertising, pay-per-use) and for software sellers and buyers. This service approach presents a striking contrast with today's stand-alone software solutions, such as ERP systems (e.g., SAP R/3), which usually require large investments in hardware and software, and an armada of well-trained personnel to deploy the system effectively. To achieve short time to market, little development is put into an overall infrastructure providing basic services and trade facilities.

Reusable components would be essential for a broader success, mainly for two reasons. First, there are many issues of data security, accounting, and transaction management that components could solve once across the various applications. Second, transparent interoperation would be a key to add value to individual services. Buyers can create their own software packages and choose the software version out of a software product line which will fit their budget. Such techniques would also facilitate the move from current aggregation software distribution to more demand- adapted ways of paying for selected data sets or for the personalized software packages [5].

These new types of pricing schemes would lower entry barriers to the market for smaller suppliers (taking into account the low set-up cost). Users would find it easier to get exactly the data they want, or the software packages (software versions) that match their budget. As the budget increases, a new data service can be chosen without need of recoding the interface to the consumers' application. In the case of stock data, for example, users can decide whether to buy expensive "premium" stock data or stick to freely available sources. They may start application development with free data, determining their needs. Once development is finished and a revenue can be derived, they will move on to a new service or provider allowing them to get an appropriate cost/revenue relationship.

2.3. Software, digital goods and information goods

Software (applications), digital goods (digital images), and information goods (digital news) are digital products that can be distributed and used in digital form over the internet. Examples are software applications, music (mp3), video (real video), text (news), images and so on. Everything that can be digitalized is either software, or a digital good, or an information good. All of these goods have a different value for customers [16].

Application service providers and intermediaries want to know how to set prices for different customer groups and how to offer digital products in sizeable software versions that are adapted to the size of the company and the budget of the consumer. Digital goods, such as mp3 music files or images, allow perfect copies to be easily created and to be distributed almost free of charge via the internet. Many software products have been aggregated (MS Office Packages), primarily to save on transaction, distribution and menu costs. But these costs are much lower on the internet. Thus, software and other types of content (data) may be increasingly disaggregated and metered, as on-demand software components and methods [8]. Conversely, buyers' main interest is to aggregate their individual software packages built up out of many software components. Every business model for providers and marketplaces of information are based on the fact that buyers differ extremely in how they value particular digital products [13].

2.4. Cost structure of digital products

For a marketplace like MMM, it is indispensable to know the characteristics and the cost structure of digital products. The cost structure of the software developer is rather extraordinary. Software, for example, is often very costly to produce but extremely cheap to reproduce. The challenge for a software provider in pricing is to find a way to sell to an audience broad enough to cover the high production costs. Development and production of a digital good involve high fixed costs and low marginal costs. The costs of producing the first copy of a digital product may be substantial, but the marginal costs of producing additional copies are negligible [16]. Large fixed costs, small incremental costs and substantial economics of scale are the very economic nature of digital products. Thus, cost-based pricing does not make any sense. Information goods should rather be priced according to the value consumers attribute to them. Still, dealing with the value of digital products, it is important to keep in mind the requirements and the personal value association of each target consumer. Consumers have widely different

values for a particular piece of information [11]. Value-based pricing evidently leads to differentiated pricing. Software producers determine a schedule of product offerings that they expect will maximize their profits. By using an intermediary like MMM, they will get more informations about the buyers' demand and therefore will be able to sharply reduce the distribution costs. In the same way, the transaction costs of administration, distribution and payment will decrease.

3. ASP vs. traditional licensing

Server-based computing is rapidly becoming the most reliable way to reduce the complexity and the total costs associated with enterprise computing. According to the Gartner Group [21], a large corporate network costs nearly $10,000 a year per seat. One third of that cost comes from hardware and software acquisition. The other two thirds come from the complexities of application installation, configuration and management. Still, most research and IT professionals have focused on a hardware-oriented view of costs. Our model presents an application-specific view. It includes how applications are deployed, the locations of users, the variety of connectivity options and the different types of client devices. We have identified the following types of costs for an ASP Marketplace:

- initial and recurring costs
- client and server hardware costs
- costs of network infrastructure
- cost of personnel
- hidden costs of lost productivity

Cost of personnel are required to develop, acquire, maintain and update applications, and provide continued technical support. Hidden costs of lost productivity are incurred when users are unable to access important applications with the appropriate level of performance. Analysts and IT professionals have developed numerous models for estimating the total cost of IT services. Most of these *total cost of ownership (TCO)* models analyze the costs of owning and maintaining a personal computer. This hardware-centric view of costs is increasingly irrelevant in the age of server based computing via the Internet. Our *TCASP* model follows an application-centric view of costs. We analyze the costs by criteria like personalized application features and functions, information logistics (delay of information), quality of user interface, location of user (PDA-version, PC-version, cellular-version) and varied types of client devices and requirements.

We identified four critical factors that determine the cost of using software applications.

1) *Physical location of the application:* The choice where an application is stored — on the server or on the client — is a determining factor in the cost and complexity of deploying and managing an application over time. IT personnel costs, as well as the time required to distribute, install and configure an application, and the cost of managing updates must be considered.
2) *Execution location of the application:* The choice of where an application actually runs, whether on the server, on the client or on some distributed combination — determines the hardware, network and connectivity required to access the application.
3) *Physical location of the data:* The choice of where the data is stored also determines the speed at which information is available, as well as the cost associated with protecting and backing up valuable corporate data.
4) *Location of the user and means of connectivity:* The user's location and network connectivity have a dramatic impact on the cost and complexity of deploying an application.

Fig. 3 classifies the three application computing models with respect to the four critical cost factors:

	Trad. Desktop	Client/Server Network	ASP Model
Application location	Client	Client or Server	Server
Application execution	Client	Client or Server	Server
Data location	Client	Client or Server	Server
User access	Local	LAN, WAN, Dial up Internet	LAN, WAN, Dial up Internet

Figure 3: Critical cost factors of computing models

As an example we will discuss application execution for each case. By using the traditional desktop model, the application is stored and executed on the desktop. This involves the need for fully functional PCs to run the application. By using a client/server and network computing model the application has to be downloaded from the server for execution on a network computer. This also involves the need for fully functional PCs on the client site to run the application. By using an ASP marketplace, applications are stored and executed on the server, thus enabling any client device to access the application. Fig. 4 shows a cost comparison of the computing models and the advantages and disadvantages of each model.

Trad. Desktop	Client/Server Network	ASP-such as MMM
Access to last and most sophisticated applications ->requires a full-function, fully configures PC risk of hardware and software failure-lost productivity sizable support staff to deploy an maintain applications highest cost of application ownership	High network traffic while downloading data and application from server Lack of network bandwidth -> loss of productivity bandwidth problematic for mobile and remote users requires still high-end client processor middle cost of application ownership	Client independent approach time and costs of installing,configuring, deploying will be reduced 'only' a personalized GUI of the application desktop is distributed to the client over any network, to any client, in any location with LAN-like performance lowest cost of application ownership

Figure 4: Advantages and disadvantages of each computing model

In order to provide access to the most recent and sophisticated applications, the traditional desktop computing model requires a full-function, fully configured PC. Also the risk of a hardware or software failure leads to a loss in productivity. As a result, the traditional desktop computing model usually leads to the highest cost of application ownership.

The client/server or network computing model generates high network traffic and leads to higher network costs. In addition, the client/server model still requires a highend client processor.

The ASP model has a client-independent approach. The time and costs of installing, configuring and deploying applications to users can be greatly reduced. MMM provides administrators with a single-point of control for deploying, managing and supporting users and applications across an enterprise network. This enables installations, updates and additions to be made only once - even across multiple servers. In addition to that only the graphical user interface (GUI) is distributed to the client. This yields the lowest cost of application ownership.

4. Business models for ASP markets

The choice of a business model represents one of the most important decisions for ASPs, as it involves the basic questions of how and how much profit should be generated. Thus, the pricing and the business model are two closely linked topics. Moreover, just as the choice of business model is a crucial issue for the ASP, it is also crucial for the ASP marketplace provider to decide which business models to *support*.

Companies may differentiate their prices following factors such as the consumers' age or time and place of consumption. Nevertheless, the pricing strategy decision is not the first one to be taken: before a company can determine its pricing strategies, it has to decide in favor of a particular business model, to which the pricing strategy will then have to be adapted. Several business models (Fig. 5) can be distinguished in the domain of media and communication [1].

ASP profit options include licensing (per user), renting, advertising, and pay-per-usage. Yet, all of these are projected to cost customers less than traditional enterprise applications. In addition to that, the different business models are linked among each other and are mostly found as a mix.

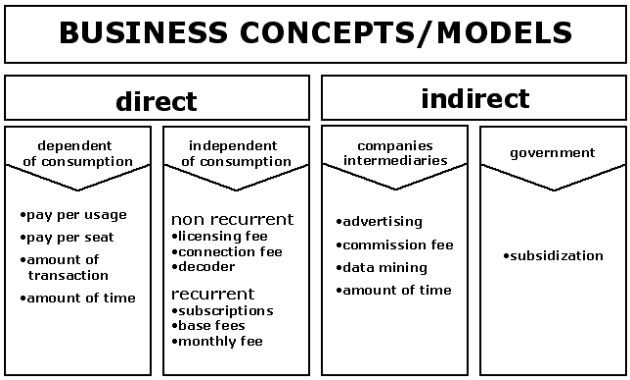

Figure 5: Business models in the domain of media and communication

First of all, we should make the distinction between direct and indirect business models. The difference concerns the presence or absence of a medium between the consumer and the supplying company. These categories can be subdivided, which eventually yields five basic types of business models.

As far as direct business models are concerned, we discern three subcategories. Two of them can be viewed as independent of the actual consumption. These payments are either non-recurrent (such as fees for connections or licensing fees), or recurrent (such as regular monthly payments in subscription contracts, or base fees). Subscription fees can be considered as a special kind of fees; their most striking characteristic is the constant amount to be paid, independent of the actual consumption.

In contrast to this sort of payment, fees may as well be calculated according to the amount of transactions supplied, or to the amount of time spent consuming (as for example at telephoning).

As stated above, the second big category of business models groups intermediate ways of payment. Here, only two subcategories can be distinguished, based on the intermediary by which the payments are executed, viz., public agencies or companies. Whereas the government intervenes by subsidies to institutions considered worth being financially sustained (e.g., granting universities free access to the internet), private companies have different incentives to mediate: not only do they attract the consumers' attention by financing their needs via advertising, but they also collect considerable information about their customers, which they can use themselves, or sell to other companies. This business model, which is based on data mining techniques [1], yields companies material and immaterial profit, as they sell information and, at the same time, use it to optimize their own operations. Furthermore, companies can play a mediating

role in transactions, and thereby get a part of the profit generated by their help. Still, these models are ideals, and therefore are not often observed in the purity we have described. They are mostly found in combinations, i.e., companies tend to adopt several business models they intend to use for different consumer groups. This enables companies and their consumers to match their interests better, maximizing profit and utility respectively, as both of them agree on making concessions. For this match to occur as quickly as possible, companies have to investigate the needs, habits and preferences of their potential customers, so that they can offer consumers the kinds of contracts they expect will best fit their budget and needs.

4.1. ASP product strategies

Especially in the domain of software, consumers tend to have very specific and absolute preferences, leading to a great variety of software component combinations to be demanded. Product differentiation is realized on the one hand through customizing and personalizing of digital products, and on the other hand through versioning.

4.1.1. Customizing and personalizing digital products

Software marketplaces and ASPs offer their customers to personalize and customize the digital product. This generates the greatest value possible to the customers. The customer must be given the choice to add or to subtract values to the software package. The most important aspect to be kept in mind in the context of this model is that the pricing arrangements capture as much of the perceived value as possible. MMM customers, for example, can use a meta search engine to describe in detail the required software component and added values like services by the means of metadata. In order for such a meta search engine to be successful, it is indispensable that software providers describe each function of a software component in a standard metadata format, preferably XML. For an intermediary it is on the other hand indispensable to know the customers' preferences. MMM offers a one-to-one communication with the customers, and thus informs the intermediary and ASPs about customers' needs and preferences. In addition to the registration, MMM uses observation functions to get demographic (zip code, age, gender) and reading behavior (queries, clickstream) information, which can be used for target advertising as a business model on MMM.

4.1.2. Versioning digital products

A great challenge for an online software marketplace and an ASP is to offer personalized products with personalized prices. Versioning means offering a product line of variations on the same underlying good [15]. The product line is designed so as to appeal to different market segments, thereby selling at a high price to those who have a high value for the product, and a low price to those who value it less. By offering product lines to customers, vendors learn about their behavior and requirements. Offering different software versions to high value customers and low value customers permits vendors to see how the market is divided. Vendors can maximize their profit by creating and developing software versions with maximum value and to sell these products by getting the highest value possible. For a software marketplace there are two principles in designing a product line: vendors must offer software versions adapted to the requirements of different types of customers; and vendors have to stress the value of each software version in a way that is transparent for the customers. There are several dimensions for versioning a software product. High value software products have a short process time guaranteed. Stock prices are of high value when customers get them in real time. The fact that customers want the latest information means that they will pay more for it. Information logistics is a way to version software components and information. High-value buyers get their information just in time, low-value buyers accept a delay. Another possibility for a software marketplace and an ASP is to provide more powerful search capacities to high value customers.

A notable feature of these dimensions is that they often involve first building the high-end software product (the immediate, high-resolution, elaborate user-interface version) and then *degrading* it in some way to produce the low-end version.

4.2. ASP pricing strategies

In order to avoid downward spiraling of prices it is important to offer a non-bulk commodity. Competition among vendors of a commodity such as digital products pushes prices towards zero [14]. Digital product commodities on the Internet, like news, roadmaps, stock prices and phone numbers, will be selling at marginal costs, that means they are for free. High first copy costs and low marginal costs form the online market structure for digital products. ASP marketplaces can offer a wide range of pricing differentiation following individual criteria so that customers' needs will be taken into account as much as possible.

4.2.1. Price differentiation

The aim for a software marketplace like MMM is to develop a differentiated product market where a couple of firms offer the same kind of digital product, but in many different varieties. This provides an alternative to a near-monopoly market, where one dominating firm offers a digital product by virtue of its size and scale of economies and enjoys cost advantages over its smaller "competitors". Online markets need price differentiation for competition. Differentiation of prices makes it necessary to differentiate the product as well. It is therefore important to develop a product line and different software versions.

These software versions are raw digital products which must be customizable according to the buyers' requirements with added value. The aim for a software marketplace and ASPs is to create a personalized and customized digital product and charge for it a price based on the value that it offers the buyer. By versioning their product, software vendors can offer their buyers to personalize and customize the product. If the software components are highly tuned to buyers' interests, vendors will have a lot of pricing flexibility. Charging each customer just what he or she is willing to pay is what economists refer to as "perfect price discrimination". In the real world, this optimum will never be achieved because it is hard to determine what is the maximum price someone will pay for a product. This restriction automatically leads to three types of differential pricing: personalized pricing, group pricing, and versioning pricing [16].

4.2.2. Personalized pricing for software marketplaces and ASPs

The main characteristic of personal pricing is the possibility of offering every customer a different price. The price each customer has to pay may depend on the quality and quantity of consumption. For MMM customers one could distinguish according to the type of enterprise (academic, small, corporate, government), the size of the organization, required databases, access time, duration time, and so on. Another option is the discount model to offer customers coupons for high revenues [14]. But by using the internet for data mining or observation, datastream vendors can make customers special offers instantaneously based on their behavior. A special form of personalized pricing are auctions. Auctions are very common and popular in the internet. Special software packages and promotion products auctions are very useful as well. Software marketplaces and ASPs should offer auctions to customers in addition to personalized pricing. Personalized pricing requires knowledge about individual customers. In MMM, for example, customers indicate their needs and the software product they would like to see, or the categories of information that are of interest to them.

4.2.3. Group pricing the classic

For online software marketplaces, group pricing (as used in classical software licensing) is not necessarily the optimum solution in the context of personalization and customization of software packages. Group pricing assumes a set of characteristics constantly valid for a whole group [7]. In some cases this makes sense, when you only know one or a few characteristics of a person (such as their age). Then you can set different prices for different groups (e.g., students). But on a marketplace like MMM you are able to figure out more than only one characteristics of a customer. With MMM it is possible to offer a personalized environment and customized products. Group pricing for a software marketplace would rarely make much sense. Group pricing makes sense for a kernel software or software product that is not customizable or personalizable. Classical software licensing models, which are based on the number of current users, number of workstations, number of servers, geographic sites, types of organization, have only one dimension for evaluation. This is not satisfying from the viewpoint of a sensible customer.

5. Conclusion

Software marketplaces and ASPs establish conditions to overcome critical application deployment and challenges of management, access, performance and security to increase organization's productivity and lower it total cost of application ownership. ASPs and ASP marketplaces are gaining momentum but their impact on software vendors is still unclear. Concerns stem from ASP models that turn one-time purchases into monthly payments. Short-term investors may not like it when the revenue spike that results from upgrading the installed base with a new release is spread over time. Still, the long-term impact is a more predictable revenue stream. Most software vendors view the subscription model as a new channel that creates incremental business by reaching smaller companies that have not been able to afford deploying their software in the past [9]. The ASP model also reduces support costs, eliminates piracy, and creates an opportunity to sell customized, personalized and complementary software and services. Start-ups are beginning to host their applications from day one. They have no installed base and no "revenue spike" issue to deal with. Why are so many software developers considering an ASP model? Vendors cite numerous incentives. First, the traditional costs of code delivery to the customer are substantial. Hosting the application could also mean that ASPs can offer faster application modifications to customers with potentially less debugging work than before. It is possible to deliver new functionality in a just-in-time manner. Recurring-revenue charge models such as those enjoyed by ISPs are also extremely appealing to developers [10]. Even installers of traditional ERP applications, who are used to large up-front application customization charges but smaller annual maintenance fees, may be attracted to the ASP model as a way to smooth revenue bumps. Finally, a web-based services model like MMM greatly expands the potential customer base for the ASP, conceivably offering access to the entire internet software business market. Furthermore, the model allows the ASP to develop its own branded services explicitly for the customer, possibly creating greater dependency and therefore greater customer lock-in.

But one of the main questions is still not answered: Are the customers and enterprises buying this service? Are

renting applications the answer to a demand resulting from end-user problems? In order to answer this question we intend to execute an empirical analysis (questionnaire) of the German software market, in a predestined branch of small and medium sized companies. Thus we want to figure out how to raise the readiness of the customers to use an ASP solution. Results of this empirical analysis should allow the carrier of software marketplaces or an ASP to design and to construct their business concepts in a way that is accepted by the consumer.

References

[1] A. Zerdick et al., *Internet-Ökonomie. Strategien für die digitale Wirtschaft*, Springer-Verlag, 1999

[2] Y. Bakos, "Towards Friction-Free Markets: The Emerging Role of Electronic Marketplaces on the Internet," *Communications of the ACM*, 41(8):35-42 (August 1998)

[3] J. Bailey and Y. Bakos, "An Exploratory Study of the Emerging Role of Electronic Intermediaries," *Int. J. Electronic Commerce*, 1(3), Spring 1997

[4] Y. Bakos, "Reducing Buyer Search Costs: Implications for Electronic Marketplaces," *Mgt. Science*, 43(12), 1997.

[5] Y. Bakos and E. Brynjolfsson, "Bundling and Competition on the Internet: Aggregation Strategies for Information Goods," Working Paper (1999)

[6] Y. Bakos, Y. and E. Brynjolfsson, "Bundling Information Goods: Pricing, Profits and Efficiency," *Mgt. Science*, 1999

[7] Y. Bakos, E. Brynjolfsson, and D. Lichtman, "Shared Information Goods," *J. Law and Economics*, April 1999.

[8] Y. Bakos and E. Brynjolfsson, "Aggregation and Disaggregation of Information Goods: Implications for Bundling, Site Licensing and Micropayment Systems,", in D. Hurley et al. (eds.), *Internet Publishing and Beyond: The Economics of Digital Information and Intellectual Property*, 1999

[9] Business Software Alliance, *Building an Inforamtion Economy*, Software Industry Positions U.S. for New Digital Era, 1999

[10] Commerce Department US government, *Emerging Digital Economy II, 1999*

[11] K. S. Corts., On the competitive effects of price-matching policies, *Int. J. Industrial Organization*, 1997.

[12] O. Günther, R. Müller, P. Schmidt, H. Bhargava, and R. Krishnan, A WWW-Based Approach for Sharing Statistical Computing Modules, *IEEE Internet Computing* 1(3), 1997.

[13] M. Dillard, *The economics of electronic commerce: A study of online and physical bookstores*. Master's thesis, UC Berkeley, 1999.

[14] A. Nevo and C. Wolfram, *Prices and coupons for breakfast cereals*, Technical report, UC Berkeley, 1999. http://emlab.Berkeley.EDU/users/nevo/.

[15] G. Shaffer, *On the incidence and variety of low-price guarantee*s, Technical report, Univ. Rochester, 1999.

[16] C. Shapiro and H. R. Varian, *Information Rules: A Strategic Guide the Network Economy*, Harvard Business School Press, Boston, 1998, http://www.inforules.com.

[17] H.-A. Jacobsen, O. Günther, G. Riessen, Component leasing on the World Wide Web, *Proc. 1st ACM Conf. Electronic Commerc*e, ACM Press, 1999. To appear in *Netnomics*.

[18] H. R. Varian, A model of sales, *American Economic Review*, 70:-659, 1980.

[19] H. R. Varian, Market Structure in the Network Age, *American Economic Review*, April 1999.

[20] H. R. Varian, Price discrimination and social welfare, *American Economic Review*, 75(4):-875, 1985

[21] L. Cohen, E. Purchase, L. Stone, and R. Terdiman., External Services Providers (ESP): Delivering on the Promise of Value, *PC Week*, June 1999

[22] U.S. Government Working Group on Electronic Commerce, *1st Annual Report*, November 1998