Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Network and Computer Applications 28 (2005) 209–232
Journal of NETWORK and COMPUTER APPLICATIONS
www.elsevier.com/locate/jnca

Application on demand system over the Internet[☆]

Ch. Bouras[a,*], A. Gkamas[a], I. Nave[b], D. Primpas[a], A. Shani[b], O. Sheory[b], K. Stamos[a], Y. Tzruya[c]

[a]*Computer Engineering and Informatics Department, Research Academic Computer Technology Institute, University of Patras, Riga Feraiou, 61 26500 Patras, Greece*
[b]*Exent Technologies Ltd, 10 Granit Street, Petach-Tikva 49125, Israel*
[c]*VP Products and Markets Strategy, Exent Technologies Ltd, 10 Granit Street, Petach-Tikva 49125, Israel*

Received 22 March 2004; received in revised form 8 June 2004; accepted 10 June 2004

Abstract

This paper describes the design and implementation of the ASP-NG system. The main modules of the ASP-NG system are the AoD service and the Web Portal. The ASP-NG Portal is a portal for providing the user with the necessary interface in order to access an Application on Demand (AoD) service. The ASP-NG portal is responsible for the interaction with the user of the AoD service. Using the AoD service the user rents an application for a limited time period at a fraction of the actual cost of the application. The AoD service is responsible for downloading the appropriate parts of the application according to the user's actions, while enforcing the mutually agreed frame between the user and the Application Service Provider (ASP). The implementation of the ASP-NG portal is based on the Web Services of the Java 2, Enterprise Edition platform and the implementation of the AoD module is based on C+ +programming language. The ASP-NG portal offers to its users the capability to select and customize the language of the user interface in order to present information in their preferred language. Moreover the ASP-NG portal offers to the portal administrator the capability to customise the look and feel of the ASP-NG portal.
© 2004 Elsevier Ltd. All rights reserved.

Keywords: J2EE; AoD; Web services; E-business; E-marketplace

☆ This work is funded by the IST (Information Society Technologies) program of EC (European Commission) under the project ASP-NG (Contract no: IST-2001-35354).

* Corresponding author. Tel.: +30-2610-960375; fax: +30-2610-960358.

E-mail addresses: bouras@cti.gr (C. Bouras), gkamas@cti.gr (A. Gkamas), inave@exent.com (I. Nave), primpas@cti.gr (D. Primpas), ashani@exent.com (A. Shani), osheory@exent.com (O. Sheory), stamos@cti.gr (K. Stamos), ytzruya@exent.com (Y. Tzruya).

1084-8045/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2004.06.003

1. Introduction

The ASP (Application Service Provision) phenomenon has swept the globe since it first appeared in the US about 2 years ago. The notion of providing software on an outsourced basis has spread to a major new trend, commonly termed ASP: The idea is that Application Service Provider delivers software applications to multiple users, provides these services for a subscription or usage-based fee and supplies these services from a central location, over the Internet or a private network, as opposed to running on the client's premises. Users could pay a monthly rental charge to an ASP (which may be the software publisher itself or a specialist service provider) who hosts and manages the software and data on a remote server. The user accesses the necessary components of the application on an on-demand basis and is authorized for each session. The user is not able to use the application outside of the mutually agreed framework with his ASP, but he experiences the application as if he actually used the complete application package and installed it in his computer. The whole procedure is handled by the AoD system and is completely transparent to the end user. Users gain access to the applications via the Internet or leased lines. The trend of ASP is increasingly extending beyond the business sector, where it is originated. Businesses are turning to ASPs because they expect different types of benefits, appealing not only to medium-large businesses with established IT departments, but also to small businesses with little or no IT staff. Expected benefits are:

- Predictable costs and low initial investments
- Enabling corporate resources to focus on mission-critical goals
- Shortening the time-to-market and time-to-benefit for new IT solutions
- Reduction of pirate software copying
- Cheaper and hassle-free use of complex packages by consumers
- New business models for software distribution

ASP-NG (ASP-New Game) project optimises the use of system resources by loading executable software to decentralised end-users on demand and to set a standard for the way applications are distributed over heterogeneous networks. The portal platform integrates all the required functionality for running an ASP operation, to deliver productivity tools and rich content and to offer secure keying of IP and Digital Rights Management. The ASP-NG project is aimed at providing ASP/ISPs, small, medium or large enterprises (SMEs and Telcos) with a complete set of features required to run an ASP. AoD enhances the usage of rich media and games protected by a unique security key, and makes use of distributed servers, such as those found in Content Delivery/Distribution Networks (CDNs).

Based on its End Users' Partners requirements ASP-NG project will concentrate on Games and Edutainment. Those application are identified by their rich content and the lack of existing solution for ASP type operations Two of ASP-NG End-Users Partners have already experience with tailor made ASP operation while the third one has Internet's Games experience. Project is dedicated to the creation of a universal platform with:

- ASP Portal
- Integrated Database

- CRM/Help Desk
- Monitoring and Diagnostics
- AoD Application Delivery

This universal platform intends to be used by small, medium and large ASP as well as within large organisation to deliver applications to external and internal users. This paper describes the design and implementation of the ASP-NG system and it focuses on its main modules, which are the ASP-NG portal, and the AoD module of the ASP-NG project. The strategic goal of the international consortium undertaking the ASP-NG project is to research, develop and implement a new middleware for Application Service Providers (ASPs), which will integrate all the required functionality for running an ASP, including new methods such as Application on Demand (AoD), to deliver productivity tools and rich content, secure keying of IP and Digital Rights Management (DRM).

The main role of the ASP-NG portal is to provide to the user the necessary interface in order to use the Application on Demand (AoD) service and interact with the business aspects of the service. The ASP-NG portal is responsible for the interaction with the end user. Through the ASP-NG portal the user is able to browse the available applications, rent and use them. In addition, the ASP-NG portal offers functionality such as searching for specific applications, checking already rented applications, creating ‘subusers’, that is, users with limited capabilities that are under the direct supervision of the parent user, credit purchase and renewal, viewing transaction history, chatting with other users of the ASP-NG system, asking questions at the ASP-NG portal’s forum, viewing already answered questions, setting parental control restrictions and more. For this reason it implements some functionality itself and some functionality is implemented in the other modules of the ASP-NG system, such as the Backoffice module and the CRM module.

Apart from the end user (‘client’) the ASP-NG portal architecture defines a number of administrative roles. The administrators are responsible for maintaining and managing parts of the AoD system. It is assumed that the administrators are going to be persons from the ASP that provides the AoD service. More particularly, the following administrators have access through the ASP-NG portal:

- *Portal Administrators*. These administrators are responsible for configuring the ASP-NG portal.
- *Content Administrators*. These administrators are responsible for application and content management (upload application metadata describing an uploaded application, create application packages, remove application from list of available applications).
- *User Administrators*. These administrators are responsible for users management (create users, edit users, remove users, switch price plans, manage relationship, handle trouble-ticketing and CRM).
- *System Administrators*. These administrators are responsible for managing the rest of the administrator users (Portal Administrators, Content Administrators, User Administrators) of the ASP-NG portal.

In addition, the ASP-NG portal offers to its users the capability to select and customize the language of the user interface in order to present information in their preferred language. When first installed, the ASP-NG portal supports only one default language (english) but the portal administrator has the capability to add support for more languages, either by translating all the strings in the portal to a new language using an intuitive web form, or by running ready scripts for languages that have already been translated. Moreover, the ASP-NG portal offers to the portal administrator the capability to customise the look and feel of the ASP-NG portal.

The AoD system by itself, consists of the following components:

- Client-side agent application
- Server-side application server
- Server-side management server
- Management console

The client side application is responsible for the user experience, allowing users to launch applications as if installed locally on the user machine. They also manage for the user the disk space and the interaction with the locally installed applications. The agent on the client-side is also used to enforce the usage restrictions that the user has subscribed to. The unique nature of the AoD system design makes sure the copyright is not infringed and that IP rights are preserved. The agent further controls the streaming and download of the application to the client machine and schedules application streaming efforts.

The application server is responsible for the secure and timely delivery of the application to the client machine. It accesses the application catalogue and the application images themselves and interfaces with an authentication and authorization database (external or internal) to authorize entitlement of users to certain application sessions.

The management server controls the application server farm ensuring load balancing and performance of the overall set of application servers. They also interact with the database to log all application session information to make sure reporting and billing is feasible.

Comparing with other AoD solutions (for example the Java Web Start), ASP-NG AoD support most of the Windows applications (including Games and Office/Education application) in contrast with the other solutions that support only a limited number of applications. For example the Java Web Start supports only applications in implemented in Java programming language.

The AoD management console is used to manage the application catalogue and the various types of parameters. The ASP-NG portal further communicates with the management database in order to properly display to users the various options they can subscribe to and to further authorize usage.

The rest of this paper is organized as follows: Section 2 presents related work that has been done in the area of Application on Demand systems. In Section 3, we present the ASP-NG system architecture. Section 4 presents the backoffice module and Section 5 presents the Application on Demand module. Section 6 describes the ASP-NG system database, while Section 7 presents the ASP-NG portal module. In addition we present some of the usage scenario of ASP-NG system in Section 8 and Section 9 presents the layout and ASP-NG web-based front-end. Finally, Section 10 concludes the paper and Section 11 discusses some of our future work.

2. Related work

Providing Applications on Demand (AoD) started around 1990 in the United States, when telecommunication, network management, content hosting and outsourcing companies pursued the ASP model. Since then there has been a systematic shift of focus to technologies based exclusively on the IP and XML standards that can be categorised in the following ASP offerings:

- Management of pre-existing applications that are based on the client/server model.
- Services through applications that are based on the client/server model, adapted to the HTML standard.
- Services through applications that are based on the IP/XML standards.

An important aspect of the implementation and a very active area of research is the proper way to implement predictive functionalities for the Application on Demand module (Highley and Reynolds, 2003). Regarding the portal interface with the end-user, the J2EE framework (Java TM 2 Platform) has been widely used in order to develop enterprise portals for demanding applications (Hazra, 2002). Its adoption has led to the introduction of competitive technologies (Miller, 2003; Williams, 2003). Our decision to develop our portal on J2EE technology was based on the wide adoption of the standard, its rich set of functionalities and the convenience of the Java language.

3. ASP-NG system architecture

Fig. 1 presents the general architecture of ASP-NG system mainly from the ASP-NG portal and AoD module point of view. As this figure shows, the ASP-NG portal interacts mainly with the Backoffice module and the users access the ASP-NG system services through the ASP-NG portal. In addition, the ASP-NG portal accesses the system database through the application server of the Backoffice module, but it has the capability to directly access the portal database. This database is specific to the ASP-NG portal module and keeps information that is related to the display languages and the different presentation skins of the ASP-NG portal. It resides within the Backoffice module, together with the system database, although it can reside in a different module, if it is desired in a specific case. The structure of the ASP-NG portal database is described in detail in Section 7.1.

The Backoffice module consists of the application server, the mail server and the database management system (DBMS). The application server provides services to other modules (the ASP-NG portal is using these services) and access to the system database. Other databases, which are module specific, are directly accessed using standard database interface (JDBC). The system database stores information related to the users, their credits, their accounts, the applications, which application has been rented by which user, etc. while the portal database stores different versions (different languages) of the ASP-NG portal, customizable interface, graphics, etc. The CRM module is responsible for handling customers' relationships. The ASP-NG portal 'drives' the end-user to the CRM through

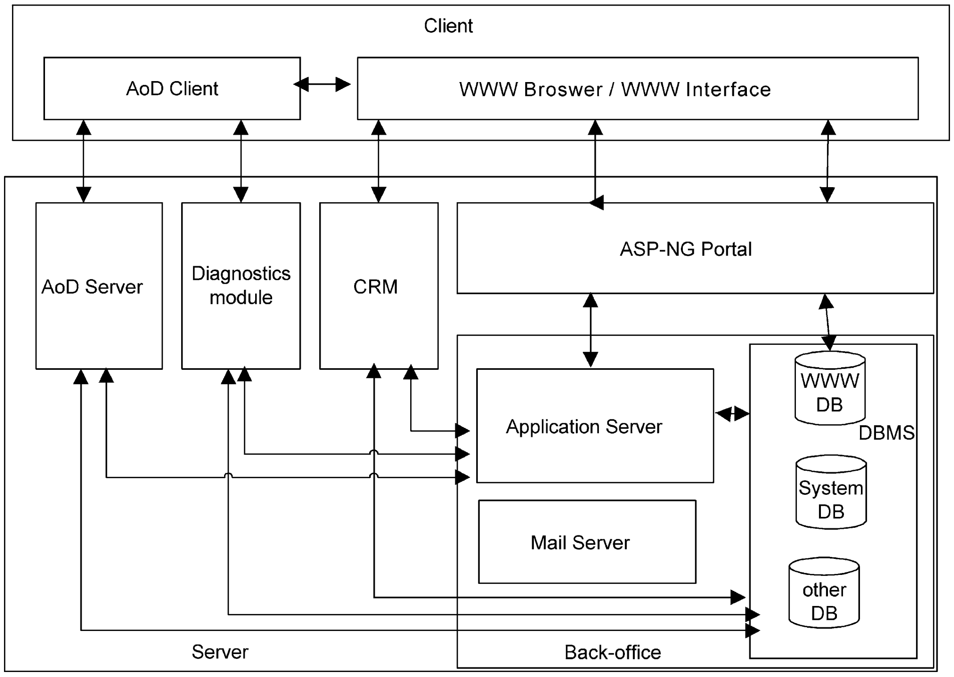


Fig. 1. General architecture of ASP-NG system.

hyperlinks for reporting problems and incidences. The AoD server is responsible for providing the core Application on Demand technology.

The ASP-NG portal interacts with the Backoffice module through the Backoffice application server. The application server provides service in API style to the ASP-NG portal. This interface is based on direct communication with the use of Enterprise JavaBeans (EJB Roman et al., 2001) technology and more particularly with the use of Java RMI, which is used by the EJB technology for distributed communication. The EJB technology offers a number of alternative methods for implementing the business logic of a web application. For this reason, the developer can choose between Entity Enterprise JavaBeans, Session Enterprise JavaBeans and Message Enterprise JavaBeans. The Backoffice module has been implemented using Entity Enterprise JavaBeans for managing data persistence in the system database. It also uses Session Enterprise JavaBeans in order to implement business logic and provide access to the Entity Beans. Therefore, the ASP-NG portal module does not have direct access to the Entity Enterprise JavaBeans, but it rather accesses the Session Beans, which construct a layer covering the Entity Beans. This technique offers a number of benefits and is called the Session Facęde (EJB Design patterns).

There is not any direct interaction between the ASP-NG portal and the CRM module except that the ASP-NG portal provides links to the CRM module. Through these links the ASP-NG portal provides to the CRM module the user ID and the user preferred language.

The diagnostic tools collect information about the status of the ASP-NG portal with the use of an SNMP agent and the SNMP protocol, which is installed in the web server of the ASP-NG portal.

The interface between the ASP-NG portal and AoD module is based on the use of Javascript technology. In addition the web browser must be compatible with ActiveX for easier interaction with the AoD client and the web server must be compatible with JavaBeans in order to access the AoD system. More particularly the interface between the ASP-NG portal and the AoD module has been implemented with the following components:

- *AoD control*. The AoD control is an ActiveX component web component that can run on the user's computer, and can be activated from the web. It exports two functions. The first retrieves some information about the user's computer and is used for example to get the installed components versions on the user's computer (such as DirectX) and compare it to the required version by an application. The second function exported from the AoD control runs the selected application on the user's computer.
- *AoD JavaBean*. The AoD module uses a JavaBean that exports functions for retrieving directions for executing and buffering an application on the user's computer. Those directions can be used as parameters for the AoD control function that runs the selected application.
- *Web and scripting Interfaces*. The web and scripting interfaces provide important AoD functionalities to the ASP-NG portal and are embedded into it. There is also a function that compares the components detected on the user's computer to the application's requirements in the database, and the following pages:
    - *Shortcut page*. Redirects an application launch request from a start menu shortcut. This way the user can start an application he has run in the past directly from his Windows desktop instead of having to browse the web site.
    - *Scheduling page*. Retrieves scheduling directions to the Player on the user machine. This functionality is useful for scheduling downloading of additional application components later.
    - *Error Log page*. Posts error events from the Player on the user machine to the client error log database table.
    - *Client Report page*. When an error occurs that cannot be solved in an automated way, the user can enter his description of the problem. The description is then submitted to this page through the HTTP protocol.

4. The Backoffice module

The implementation of the Backoffice API module is based on the J2EE (Java 2, Enterprise Edition (Java TM 2 Platform) and the implementation platform selected is the Oracle Application Server 9i Java Edition (Oracle AS9i) (Oracle corporation). Oracle AS9i is one of the most advanced and reliable J2EE compliant application servers available today. The implementation of the Backoffice API module does not rely on any

Oracle AS9i proprietary features and a pure J2EE application has been implemented. This means the Backoffice API module can be deployed and run on any J2EE compliant application server without (or with minimal) changes. On the development phase, the OC4J (Oracle Containers 4 Java) was used for deployment and testing of the Backoffice API module for simplicity reasons. For actual deployment the Backoffice API module is designed to be deployed on the full Oracle Application Server 9i. The difference between the OC4J and full Oracle Application Server 9i is that the Oracle Applications Server 9i contains more modules besides the OC4J, like the HTTP Apache server for front-end, the Web Cache module etc.

The Backoffice API module consists of entity and session EJBs. In order to make the development of the Backoffice easier, the Oracle JDeveloper programming environment was used, as it provides an advanced Integrated Development Environment (IDE). The Entity beans are implemented as bean-managed EJBs, which means that all the load, store, insert methods etc. were implemented by code written by the programmer (and not automatically generated by the container). The alternative, of using the capability of container managed EJBs, where the EJB container has the responsibility for programming these operations was not used in the ASP-NG implementation because of the immaturity of the available Container-Managed Persistence (CMP) implementations (Fig. 2).

On the other hand, the session EJBs are implemented according to the Backoffice API module design and they are stateless session EJBs that expose their methods in local

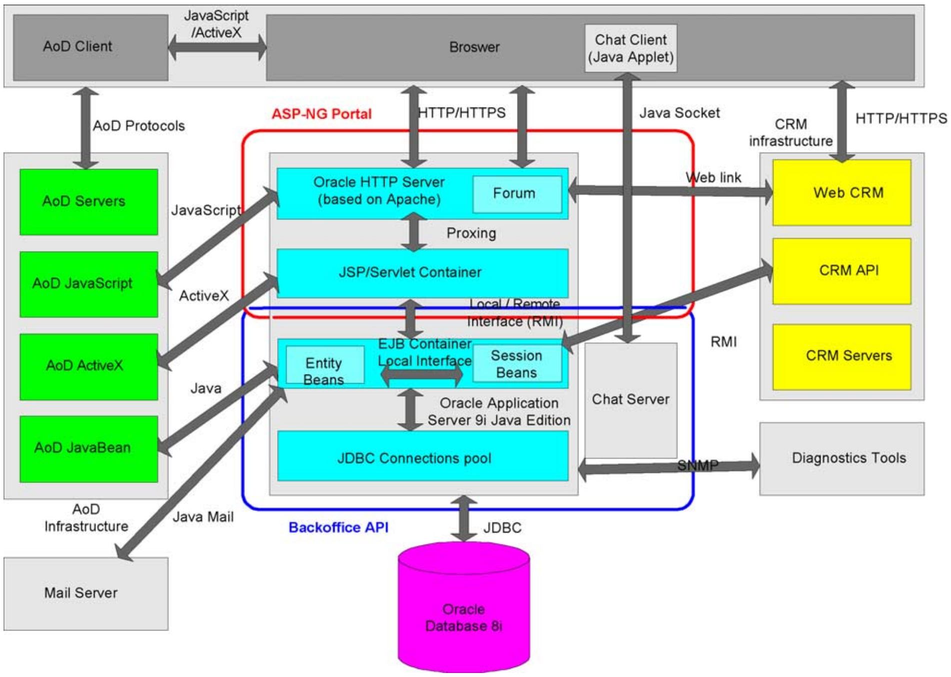


Fig. 2. ASP-NG BackOffice API implementation technologies.

and remote interfaces. Generally, in the implementation of the Backoffice API both local and remote interfaces are provided, so that in the future it can be possible to change the communication between session and entity EJBs with minimum modifications on the source code.

In addition, each entity bean corresponds to a table on the ASP-NG system database. For the communication between the entity EJBs with the tables, a JDBC connection is used. The JDBC connection can be established using JBDC driver and the JDBC API that allows to access the database using pure Java programming language.

5. The application on demand module

The architecture of the AoD module is shown to Fig. 3 and consists of the following sub-modules:

- *Content Encoding*. The Content Encoding enables the creation of new contents for the ASP-NG based on the original CDs.
- *DRM and Key Security*. This mechanism is used by the client software in order to verify that only users that are authorized to access the application are able to execute it.

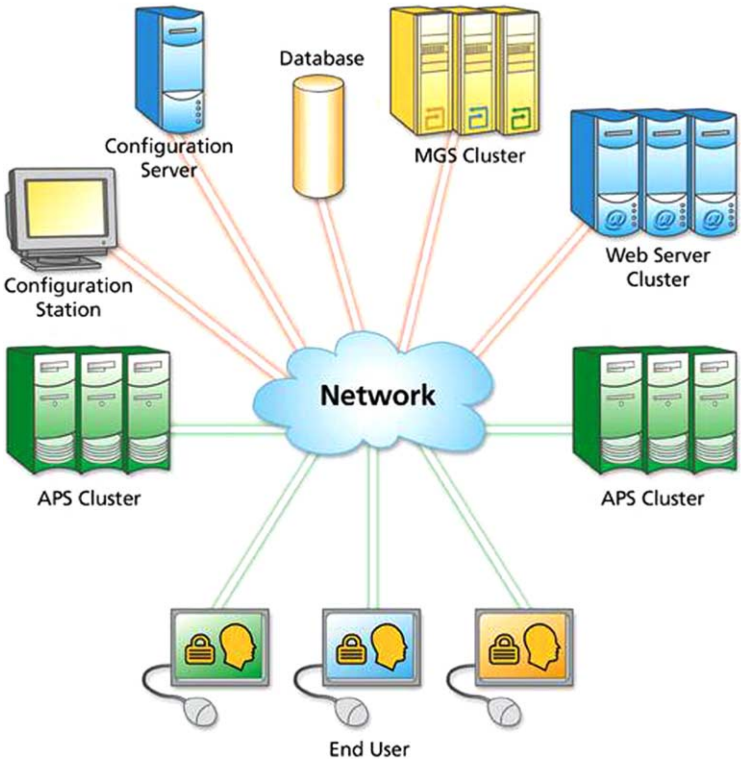


Fig. 3. AoD module architecture.

- *Content Delivery*. The control of the process of content delivery.
- *Prediction Server*. The Prediction Server gets prediction logs of the usage of applications and translates them to prediction scripts used by the client in order to improve streaming performance.
- *Managed use of the application*. The client-side agent controls the use of the application, enforcing authorization restrictions and interfacing with the other components to make sure the user experience is complete.

5.1. Content encoding and distribution

The goal of the encoding phase is to prepare the application for streamed, online delivery mode. The ASP-NG goal is to enable the user to start using the application as quickly as possible. As such, the AoD client can start running the application even if it does not completely reside on the client machine. In order to identify the parts that are needed in order to start running the application and to keep being ahead of the user activities, streaming the parts of the application the user will need next, the encoding process analyses the application use and creates prediction maps.

One of the most important issues of maintaining the AoD servers is the content distribution and management. Since the AoD system is a distributed system, the content distribution process is complicated. The content should be copied first to an APS and then all other component (e.g. web, Direction Supplier) should get the relevant information (‘master record’) about this content. Using an automated process for most of the process’s steps simplifies the process. The Distribution tools should have the ability to be integrated to existed distributed systems. Sharing one content between several APSs can simplify the content distribution process. Any content should be copied only once and the integrity test should be done only once.

5.2. DRM and key security

The online digital rights management system role is to protect the content provider’s intellectual property by encrypting the data so that only authorized users can access it. In order to execute an application, the end user accesses the ASP-NG portal and choose the application, the BackOffice server then authenticates the user (e.g. by username and password) and only then it starts a new session by generating a ‘Ticket’ and sends it to the client in the direction block.

The ticket is passed from the end user’s browser to the AoD Player, which sends it to the APS. The APS decrypts the ticket and only if the ticket is valid it allows the user to read the relevant content. The application can be executed only through the AoD Player and only when the user is connected (‘online’) to the APS. If the AoD Player disconnects from the network for a long time it will stop the execution.

The ticket is encrypted by the web and decrypted in the APS. A shared key is written in the AoD database and distributed to the required system’s components.

Authentication mechanism that is based on username and password is sometimes not strong enough. For example, business models that are based on 'user subscription' allow the user to rent an application for a period of time and to play as much as he wants in this period. The user can share his identification with other people and to play more then one session simultaneously, while having paid only once. In order to prevent users from playing simultaneously more than the number of sessions they have paid for, a stronger authentication mechanism is required.

A lot of applications have their own DRM solution which most of the time is implemented by typing a CD key or an activation key at installation time. This key is stored in the computer registry or in a file in the file system. The application authenticates the user as the application's owner and validates this key.

The application's activation key can be depending on some parameters in the end user computer (e.g. MAC address, drive C serial number etc). In order to support execution of those applications, the AoD system uses the CD key server to generate an activation key; supply the key to the end user in the Direction.

5.2.1. Overview of online-session ticket creation

When the end-user starts a session, he connects to the ASP-NG portal and chooses a game. The web generates a new Direction, which contains the ticket block. A ticket is encrypted buffer, which contains some parameters about the user, the required application the session and time limitation for this ticket.

Each ticket contains two time parameters:

- *Issue time*. The time the ticket has been created.
- *Expired time*. The time this ticket should be expired and the AoD Player should prevent execution of application with this ticket.

The ticket mechanism is based on the following principals:

- Each ticket contains time limitation ('Expired time'). When this time passed, the APS disconnect the user and the session is closed with appropriate logging in AoD database.
- Each ticket is valid only for defined period of time; the user cannot use this ticket if the time period passed.
- Each ticket can be used only once.

5.2.1.1. Ticket encryption. The ticket is generated and encrypted in the web by a java bean that has been developed for this purpose. It is being decrypted in the APS server that the AoD Player connects to. The AoD Player itself does not read or change the information stored in the ticket and it does not have the key to open the encryption.

The crypt key is shared between the APS and the Java bean. It is located in the AoD database. The java Bean can read it directly from there, the APS reads it through secured connection to the MGS. The Key can be changed by the provider and should be different in each system.

5.2.2. Scheduling protection

When the end-user starts a scheduling session, the AoD Player connects to the ASP-NG portal to get a new Direction. Since the AoD Player has to access this web page without any user intervention this web page is not password protected. This may lead to a security breach because unregistered hackers, that have broken our protection algorithm, can access this unprotected web page, and use it to get Direction. To solve this security breach the Direction and the Ticket that this web page creates can only be used in order to preload the application, and not to run it. The APS is responsible for validating this ticket and for preventing the session that uses this ticket to start running the application.

5.3. Prediction capabilities

An application consists of files. Part of the files are required every time the application is launched or at start up time and part of them are required only if the end-user uses a specific part of the application. For example, in games the executable is always required, sometimes the application starts with a short movie, which is required at start up but the application's scenarios are required only when the user reaches each one of them.

The AoD player first downloads the application parts that are required for launching the application. When this step is finished, the user can launch the application. The AoD player continues to read other application components from the APS in the background. When the application asks for some data, which is not cached locally, the AoD player reads it from the APS. This operation can take some time. The AoD Player uses a 'read ahead' mechanism that reads the required files/blocks from the APS server before the application requires them. This mechanism increases the 'hit ratio' and reduces the time the application waits for blocks to be retrieved from the APS. This improves the quality of the service by improving the end-user's experience.

As mentioned, a 'Prediction File' is created for the applications during the encoding process. The decision, which portions from which files will be downloaded and what is the right order, is not a dichotomy. Different users use the same application in different ways and different blocks/files are required (e.g. in game user can chose one of two different scenarios).

In order to predict which blocks to read, and to improve our 'hit ratio', the AoD Player builds a 'prediction thread' from a list of all blocks that have been accessed by the application. The 'prediction thread' is being saved in a 'prediction log file', which being sent to the prediction server by the AoD Player when the application terminates.

6. The ASP-NG system database

The Backoffice module manages the ASP-NG system database, which stores the users, the applications, the rental objects and all their relationships. In addition, the database also contains an independent table that is used to keep some settings parameters that can only be changed by the ASP-NG system administrators. At the first installation of the ASP-NG system this database is mostly empty and the content administrator is responsible for

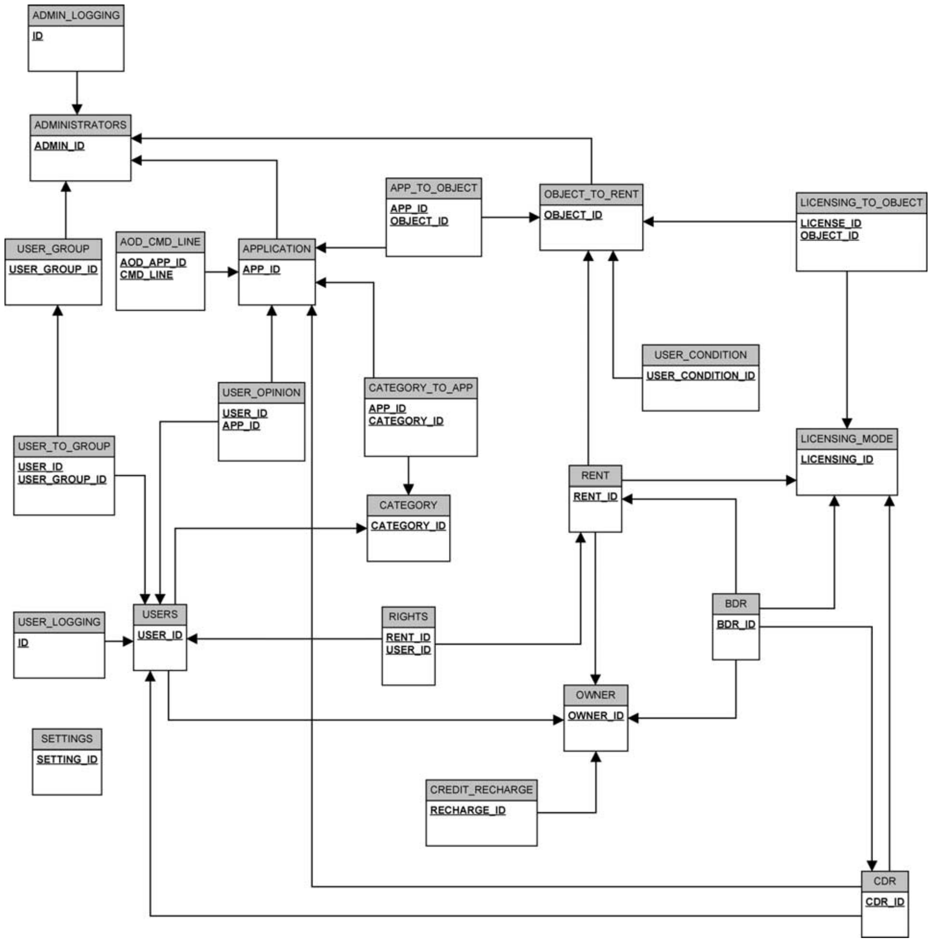


Fig. 4. ASP-NG system database schema.

inserting the available applications for rent. However, the database will have some initial values on the settings table that are inserted at the installation phase of the ASP-NG portal. The system administrator must check them very carefully before any other action takes place on the ASP-NG system. Fig. 4 shows the structure of the system database.

7. ASP-NG portal module

The implementation of the ASP-NG portal has been based on the Web Services of the Java 2, Enterprise Edition (J2EE) platform (Java TM 2 Platform). More specifically, the ASP-NG system makes use of the JavaServer Pages (JSP), Servlet and Enterprise JavaBeans (EJB) technologies.

The J2EE technology provides a component-based approach to the design, development, assembly and deployment of enterprise applications, in order to reduce costs and enable faster enterprise application design and development. Its multitier distributed application model is not tied to the products and Application Programming Interfaces (API) of any one vendor. J2EE applications are made up of components. A J2EE component is a self-contained functional software unit that is assembled into a J2EE application with its related classes and files and that communicates with other components. The web services of J2EE specification defines client components, Servlet and JavaServer pages components that run on the server, and Enterprise JavaBeans components that run inside the EJB container.

Fig. 5 shows the J2EE Web Services Architecture. In the ASP-NG system the server-side presentation module has been implemented as part of the ASP-NG portal and the server-side business module has been implemented as part of the application server of the Backoffice module. The architecture of the ASP-NG portal has been based on JavaServer pages that mainly contain the presentation data for the ASP-NG portal. In order to separate business logic from presentation logic, the JavaServer pages use functionality, which has been implemented inside Enterprise JavaBeans, hosted by the Backoffice module. Using this approach we can also benefit from the transaction, security, performance and stability features offered by the EJB container of the Application Server. The JSP pages do not directly access the main ASP-NG database. Instead, they use Enterprise JavaBeans methods that enable the JSP pages to access the data they need. The ASP-NG portal also uses the internationalisation features offered by the Java technology in order to implement the multilingual functionality of the ASP-NG portal.

Customization features of the ASP-NG portal are partly implemented using the Cascading Style Sheet mechanism. The Cascading Style Sheet mechanism allows authors and readers to attach style such as fonts, colors and spacing to HTML documents, using a human readable language. The ASP-NG portal stores a different stylesheet for each different skin that the portal administrator installs. The contents of the selected (by the portal administrator) stylesheet are requested from the user through an intuitive web form. From then on, each time this user requests any page from the ASP-NG portal, the system will provide the requested page with a link to stylesheet included in the HTML ⟨link⟩ tag.

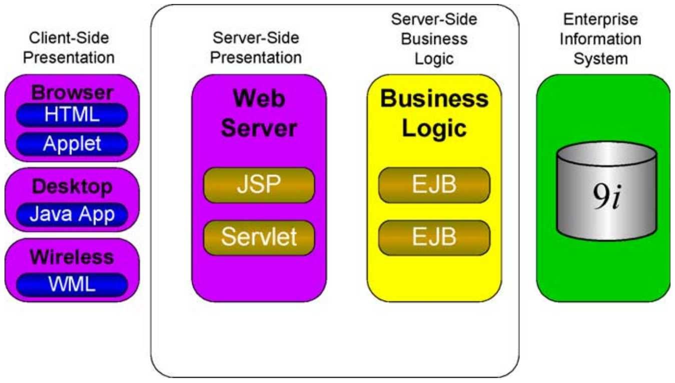


Fig. 5. J2EE Web services architecture. (Source of figure: Oracle Technology Network (http://otn.oracle.com).

7.1. The ASP-NG portal database

The ASP-NG portal database supports the operation of the localization of the customized look and feel of the ASP-NG portal. In order to support that service the ASP-NG portal database contains eight tables. Fig. 6 shows the ASP-NG portal tables and their relationships.

When the portal administrator wants to insert a new language to the ASP-NG portal, the necessary information is inserted in the Languages table. The portal administrator can then select the page to translate the strings contained in that specific page for the new language. He is then presented with a form where he can translate all strings contained in the selected page. If some string has already been translated, the corresponding edit box is already completed. With the Database tables completed (through the WWW forms that the portal administrator can view), the.properties files can be generated. A similar approach is applied in order to add new skins and create the necessary.properties files.

For the correct operation of the ASP-NG portal, during the installation of the ASP-NG system, the portal database contains (at least) all the necessary information for the operation in the default language (English) and the default skin. After the installation, the portal administrator has the capability to add new languages and new skins. Because the default language and the default skin are used as guidelines for adding new languages and new skins, it is the responsibility of the portal administrator to check that the new default skin and default language are properly installed, when he changes the default language and the default skin.

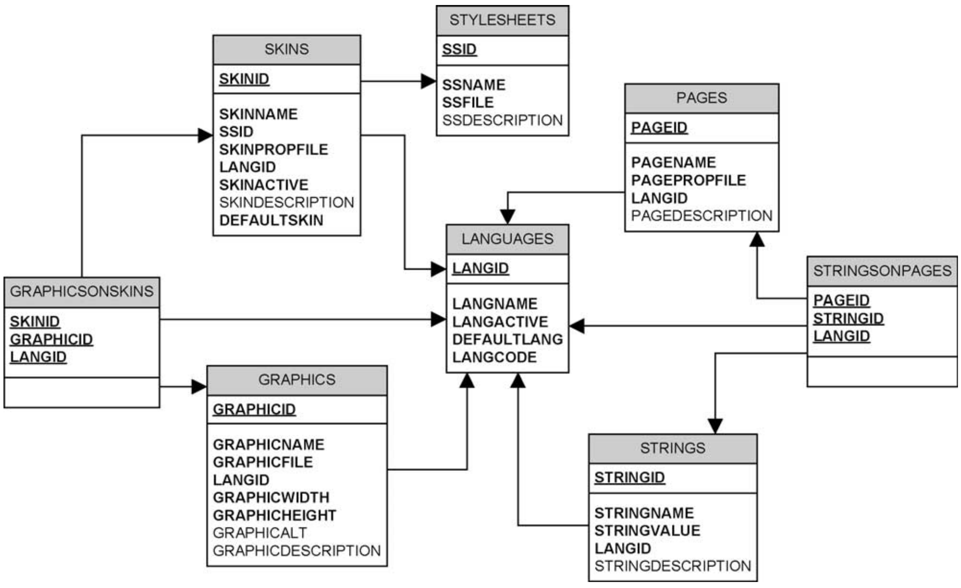


Fig. 6. ASP-NG portal database schema.

7.2. Authentication and security issues

The ASP-NG portal offers a built-in authentication system for registering and validating registered users. The authentication of a user is based on a login/password combination over secure HTTPS/SSL (The SSL Protocol) connections and the users' passwords are stored encrypted in the ASP-NG Backoffice database. With this approach even the system administrator does not have the capability to read a user password (of course the system administrator has the capability to reset the user password). We have decided to base the authentication mechanism on the common approach of login/password combination and not to use more advanced mechanisms (for example user authentication with the use of certificates) for the following reasons:

- The proposed authentication system (login/password combination over secure HTTPS/SSL connections) provides a good security level, which satisfies the requirements of an e-commerce solution like the ASP-NG service.
- Most of the end-users are familiar with the use of login and passwords.
- A more advanced authentication system, which needs specific technical knowledge from the user, adds overhead to the authentication mechanism and may discourage a user without the necessary technical knowledge.

In addition, the ASP-NG portal provides the capability of secure communication in the means of secure HTTP/SSL connections. Every time the user provides personal information and passwords (for example during registration or authentication) or during renting transactions (for example in order to rent an application) all the communication is done over secure HTTP/SSL connections. We selected the approach to provide only the most sensitive parts of ASP-NG portal over secure HTTPS/SSL connections and not the whole ASP-NG portal, in order to increase the performance of the ASP-NG portal, due to the fact that the HTTPS/SSL connection includes an overhead to the overall performance of the ASP-NG portal, since it has to encrypt the whole content of all pages protected by HTTPS/SSL.

7.3. ASP-NG portal localization features

The ASP-NG portal offers to its users the capability to select and customize the language of the user interface in order to present information in their preferred language. Initially the ASP-NG portal supports only a small set of languages (during the developing phase these languages were English and Greek) but the portal administrator has the capability to add support for more languages at any time, through a very simple web-based interface. The implementation of the ASP-NG portal multilingual user interface is based on the internationalisation features of the J2EE platform. The J2EE platform provides a rich set of APIs for developing internationalised WWW portals like the locales (class java.util.locale) and resource bundles (class java.util.ResourceBundle).

On the J2EE platform, a locale is simply an identifier for a particular combination of language and region (for example language English and region USA). Java locales act as requests for certain behaviour from another object. This object tries to fulfil the requested locale but if this is not possible, it then selects the best available locale (for example USA English instead of UK English) or if there is not any matching locale then the object uses the default locale. The process of selecting the best matching locale for a specific locale request is handled by the J2EE core classes and there is no need for special programming. The identification of users' preferences regarding localization is based on ISO 639 for languages identification and ISO 3166 for country identification.

Resource Bundles contain locale-specific objects, which are in most of the cases locale-specific strings. When a program needs a locale-specific resource, for example a string, the program can load it from the resource bundle that is appropriate for the current user's locale. With this approach, the programmer can write code that is largely independent of the user's locale isolating most, if not all, of the locale-specific information in resource bundles. A Resource bundle obtains locale specific resources from properly formatted text files (with the extension '.properties') or properly defined java classes.

More specifically, for the implementation of the ASP-NG portal multilingual user interface we have used locales for identifying the user language preference either indirectly by the user browser setting or directly by asking him for his language preferences. The only information we are interested in for each locale is the locale's language (and not the country or the specific variation of a language, since we differentiate by language selection). The information for each different language of the ASP-NG portal is stored in the portal database and every time an interface item (for example a string) changes or every time the portal administrator decides, the appropriate resources bundle files (.properties files) are created based on the portal database information. With the above approach—building the resources bundle files on demand and not extracting the information of the portal database during the operation of the ASP-NG portal—we improve the performance of the ASP-NG portal. In order to generate a web page and serve it to the user's browser, the server does not have to make any additional database access. It only reads a local.properties file, which contains the strings of the specific page, translated in the suitable (preferred) language. In addition, the fact that the localization information is not changed very often leads us to the above-described approach.

7.4. ASP-NG customised look and feel

The ASP-NG portal offers to the portal administrator the capability to customise the look and feel of the ASP-NG portal. With the term 'customise' we mean the capability that the portal administrator has to perform the following operations:

- Change (create, edit, remove) the stylesheet (.css file), which controls the look and feel (for example fonts and colour of each element) of the ASP-NG portal.
- Change (create, replace, remove) the graphic files of the ASP-NG portal with others with the same dimension.
- Change the strings of the ASP-NG portal (this capability is provided through the localization features of the ASP-NG portal).

The realisation of the above features of the ASP-NG portal has been based on the concept of Skins (Geary, 2001). Skins are a common approach to provide customised look and feel not only to WWW portals but also to other kinds of applications. With the term 'Skin' we refer to a collection of interfaces objects (for example graphics, font sizes, font colours, etc.), which can be used to change the user graphic interface. In the case of the ASP-NG portal the skins consist of the portal stylesheet and the portal graphics. For the implementation of the customised look and feel feature of the ASP-NG portal the concept of the Skin Server is used. The Skin Server is a program (in the ASP-NG portal it is a Servlet program), which is responsible for loading the skins information and providing the correct object for each request of a skin element based on the ASP-NG portal configuration. The SkinServer provides its services to the other JSP pages of the ASP-NG portal. The skin information is stored in the portal database and based on that information the portal administrator can build special formatted files (.properties files) with the skin information. We choose this approach—similar to the one implemented for the localization features—for better performance as we have already explained in the previous paragraph.

7.5. ASP-NG portal implementation technologies

The ASP-NG portal is responsible for the interaction between the system and its clients. The main service of the ASP-NG portal is the access to the Application on Demand (AoD) service that the ASP-NG system offers. In addition to that service, the ASP-NG portal offers a number of supportive services, which increase the effectiveness, and the user-friendliness of the ASP-NG system. The most important supportive service is the usage of the user profiles. Each user profile contains information regarding the end user, his/her authorities, statistical information, billing information etc. In addition the ASP-NG portal offers customized look and feel capabilities and internationalisation.

The security and authentication features of the ASP-NG portal use state of the art technologies and protocols in order to ensure the secure and proper operation of the system. These technologies and protocols include secure protocols like the HTTPS (Secure HTTP) SSL (Secure Socket Layer) and public/private key technologies. The ASP-NG portal is implemented with the use of JSP and Servlets technologies, and EJBs for implementing business logic with regard to localization and customization features.

The Backoffice is implemented using J2EE technologies and more particularly using Enterprise Java Beans (EJB) technologies. We use the EJB version 2.0 standard and the following types of EJBs defined in EJB version 2.0 specifications:

- Entity EJBs—mainly BMP (Bean Managed Persistence) Entity EJBs: For modelling the ASP-NG system database in the Java Environment.
- Session EJBs (both stateless and statefull). For providing the Backoffice API calls to the Backoffice users (JSP pages of ASP-NG portal). The session EJBs use the entity EJBs, which represent the System Database in the Java Environment.

The Backoffice API EJBs are used by the ASP-NG portal using the following concept. The Backoffice API provides the business logic to the ASP-NG portal and the ASP-NG portal provides the presentation layer to the end user.

The Backoffice API EJBs are deployed to the application server of the Backoffice and the communication between the entity EJBs and the session EJBs is based on Local Interfaces (running in the same Java Virtual Machine—for performance reasons).

The communication between the session EJBs and the ASP-NG portal is based either on Local Interfaces (both Backoffice EJBs and ASP-NG portal running in the same Java Virtual Machine/Container and the same physical server in order to achieve better performance) or on Remote Interfaces (the Backoffice EJBs and ASP-NG portal running on different physical servers in order to achieve better scalability).

8. Usage scenario of ASP-NG system

The ASP-NG system front end consists of the ASP-NG portal, which is a generic web site designed for AoD implementation. The ASP-NG portal's front-end interface includes a comprehensive set of features designed to facilitate rental uptake and to enhance user experiences with an Application on Demand service. The user accesses the AoD service through the ASP-NG portal. The user has the capability to browse the ASP-NG portal and collect information about the available applications in the AoD server. The ASP-NG portal offers to the user different ways of browsing (for example browse by category or browse all applications) and searching (for example searching by keyword, searching by category etc).

In order to rent and use an application the user must be registered in the ASP-NG portal. A registered user has to submit his login and password information in order to be authenticated. After passing the authentication control and if the user has enough credits in his virtual account, he can rent and use applications. The ASP-NG system offers to the user many capabilities for renting an application (rent one application, rent a bundle of applications, etc.). The system also simulates a billing system by offering to the user the capability to buy credits in different ways, like paying by credit card, paying through a bank account, paying directly or paying using a micro payment mechanism. In addition, after the user has passed the authentication control, he has the capability to update his profile, which contains all the user-defined details of the user's account. Through the profile the user has the capability to access and update information relative to his account such as his personal information, his transaction history, and his system preferences. In addition, the user has the capability to manage (create, edit, remove) sub-accounts for subusers with limited access to the applications rented by the parent user. The level of access for the subusers is determined by the parent user. This way, the parent user is able to create for example accounts for his children by restricting access to applications that the parent finds unsuitable for their age. The sub-accounts only have access to the applications through the credits of their parent account (cannot rent their own applications).

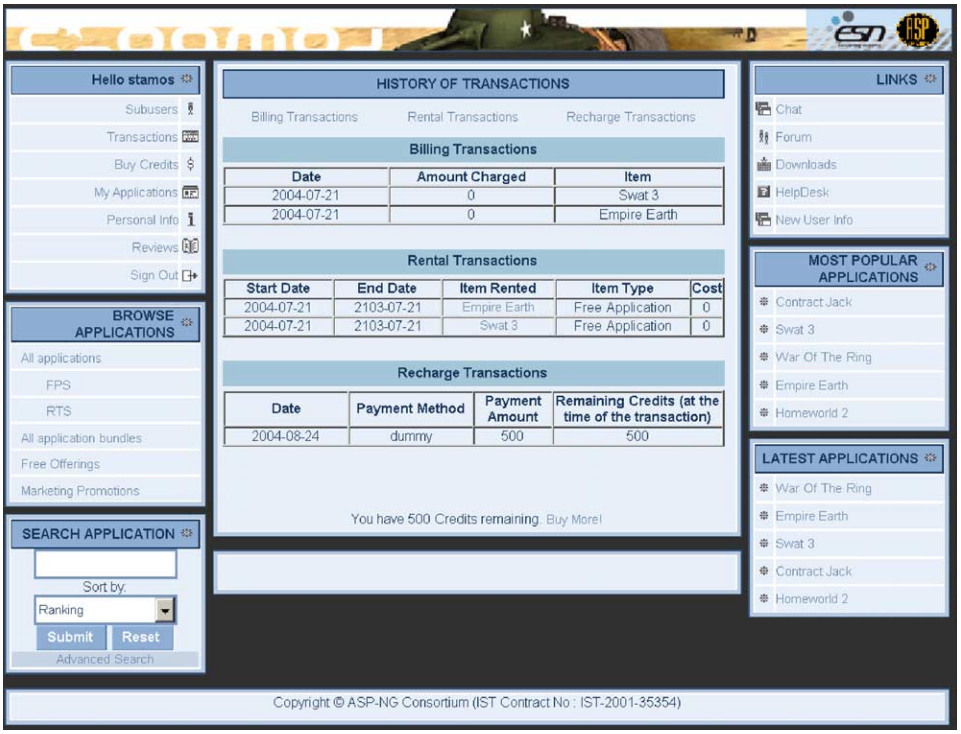
Moreover, the ASP-NG portal offers several methods for communication among its users. For example the ASP-NG portal offers chat rooms (general chat rooms or chat rooms with specific subjects) and forums (general forum or forums with specific subjects). When a problem occurs or if the user has any questions regarding the operation of the system, the ASP-NG portal guides the user to the Helpdesk/CRM module (Customers Relationships Management) through which the end-user has the capability to access help pages (for example guidelines, FAQ) or is able to contact Helpdesk personnel.

A registered user can run an application he has rented either by visiting the ASP-NG portal and selecting the appropriate application, or click on a shortcut that is automatically created in his Windows shortcuts after the first time he uses an application. Once the user starts an application, the portal module is responsible for creating an encrypted one-time ticket for this session of the user, which will be passed to the AoD module in order to certify that this user is allowed to run the application for a certain time period. This procedure is described in detail in Section 5.2.1.

9. Layout and ASP-NG web-based front-end

Fig. 7 shows the general layout of the ASP-NG portal and more particularly the layout of the logged user. The logged user page consists of the following:

- *Advertisement (Top)*. Displays an advertisement banner. The contents of the banner are determined by the portal administrator, who can for example use this space for commercial or promotional purposes.
- *User Login or User profile area (Upper left)*. There is one link about the user's subusers, which leads to a page that displays all subusers this user has created. Another link leads to the user's transaction history. The transaction history page displays all purchases and rentals this user has made. The next link leads to this page, which lists the applications that the user has rented. The following link leads to a page where the user can see and edit his personal information and preferences, and the last link leads to



HISTORY OF TRANSACTIONS

Billing Transactions Rental Transactions Recharge Transactions

Billing Transactions

Date	Amount Charged	Item
2004-07-21	0	Swat 3
2004-07-21	0	Empire Earth

Rental Transactions

Start Date	End Date	Item Rented	Item Type	Cost
2004-07-21	2103-07-21	Empire Earth	Free Application	0
2004-07-21	2103-07-21	Swat 3	Free Application	0

Recharge Transactions

Date	Payment Method	Payment Amount	Remaining Credits (at the time of the transaction)
2004-08-24	dummy	500	500

You have 500 Credits remaining. Buy More!

Copyright © ASP-NG Consortium (IST Contract No : IST-2001-35354)

Fig. 7. Logged user home page.

the page where the user can view and edit his ranks/reviews for applications he has tried and wants other users to know his opinion on them.

- *Browse area (Middle left)*. Here the user can browse the applications offered by the ASP-NG system. He can either select a specific application category, or view all individual applications, all application bundles, or all products offered by the system.
- *Search area (Lower left)*. the user enters the keywords in the text box and chooses the sorting criteria (product rank, product category). There is also a link to the advanced search page. The advanced search page offers many more possibilities for the user to enhance his search and achieve more focused and relevant results.
- *Main area*. There is a list of the applications and application bundles this user has rented and a link for ranking them, and a line showing the user's remaining credits and offering the possibility of buying more. The user's bandwidth has also been measured and is reported at the lower part of the page. The user has also the possibility to perform the bandwidth measurement again. The user can also select to view his most recently used applications.
- *Collaboration area (Upper right)*. Links to the chat, forum pages and CRM functionality.
- *Top Titles area (Middle right)*. The names of the top 10 applications are shown in this area, each one with a link to more applications details.
- *New Titles area (Lower right)*. Here the names of the latest applications added to the ASP-NG portal are displayed, each one with a link to more application information and details.
- *Featured application (Bottom)*. This area allows the promotion of one application, presenting some details and perhaps graphics. If the portal administrator chooses not to offer some featured product, this page area is removed.

10. Conclusion

In this paper we presented the design and implementation of the ASP-NG system, which constitutes an integrated system in order to use AoD service. The ASP-NG portal, which is the web interface for accessing the AoD service, is based on current state of the art J2EE technologies and offers enhanced capabilities like multilingual interface and customised look and feel. The design of the ASP-NG portal offers enhanced performance and scalability based on J2EE technologies, separates the presentation and the business logic and allowing these two entities to operate independently. The AoD module, which is the core of the ASP service, is equipped with all the necessary tools for administering the content encoding, scheduling and distribution mechanisms.

11. Future work

Our future work includes the evaluation of the ASP-NG system through its actual deployment at three different application providers' sites, in the context of the ASP-NG project. We plan to gather the observations and results that will be produced through the exhaustive usage of the ASP-NG system by hundreds of end-users and then possibly use them to enhance the current implementation. More particularly during the Pilots, the different partners will provide support and management assistance in order to solve issues and provide a stable online service. The pilots will include testing of all the subsystems designed within the project. The pilots shall include at least 300 users each, using the profiles defined by the market survey. Finally, their duration, with actual users, will be for at least 3 months.

Acknowledgements

We want to thank the European Commission (EC) for supporting this work in the context of ASP-NG IST project (Contract number: IST-2001-35354). In addition, we want to thank our partners in the ASP-NG project for their valuable cooperation. More information regarding ASP-NG project can be found in http://www.asp-ng.eu.org

References

EJB Design patterns, advanced patterns, processes and idioms, Floyd Marinescu Wiley Computer Publishing (ISBN 0-471-20831-0).

Geary D. Advanced JavaServer. New Jersey: Prentice Hall; 2001. PTR; ISBN: 0130307041.

Hazra T. Building enterprise portals: principles to practice. Int Conf Software Eng, Orlando, Florida 2002;623–33.

Highley T, Reynolds P. Marginal cost-benefit analysis for predictive file prefetching. Proceedings of the 41st Annual ACM Southeast Conference (ACMSE 2003), Savannah, GA; March 2003.

Java TM 2 Platform, Enterprise Edition (J2EE), http://java.sun.com/j2ee/

Java Web Start: http://java.sun.com/products/javawebstart/

Miller G. The Web services debate: .NET vs. J2EE. Commun ACM 2003;46(6):64–7.

Oracle corporation: http://www.oracle.com

Roman, Ambler SW, Jewell T, Marinescu F, editors. Mastering Enterprise JavaBeans. New York: Wiley; 2001. ISBN: 0471417114.

The SSL Protocol http://wp.netscape.com/eng/ssl3/draft302.txt

Williams J. The Web services debate: J2EE vs. .NET. Commun ACM 2003;46(6):58–63.

Further Reading

Ambrose C. Preparing for application service provider implementations, Gartner Commentary COM-16-0871: Gartner Inc; 2002.

Bouras C, Gkamas A, Primpas D, Stamos K. Application on Demand Portal using J2EE technologies. IADIS International Conference—Applied Computing, Lisbon, Portugal; March 23–26 2004, p. I.551–I.558.

Lucas U, Schwabe U, Wojcik L. Synchronous software on demand over the internet—opportunities for end users, software manufacturers and online providers. ECEC 2000: Concurrent Engineering in the Framework of IT Convergence; April 2000.

Christos Bouras obtained his Diploma and PhD from the Computer Science and Engineering Department of Patras University (Greece). He is currently an Associate Professor in the above department. Also he is a scientific advisor of Research Unit 6 in Research Academic Computer Technology Institute (CTI), Patras, Greece.

Apostolos Gkamas obtained his Diploma, Master Degree and PhD from the Computer Engineering and Informatics Department of Patras University (Greece). He is currently an R&D Computer Engineer at the Research Unit 6 of the Computer Technology Institute, Patras, Greece. His research interests include Computer Networks, Telematics, Distributed Systems, Multimedia and Hypermedia.

Itay Nave is Chief Technology Officer at Exent Technologies Ltd. As CTO, Itay Nave leads the development of the EXEtender™ platform, and related services. Nave is responsible for materializing Exent's product and technology strategy, affirming and maintaining the company's position as a dominant player in the industry.
Nave produced the basis for Exent's Applications-on-Demand systems while he was a project manager in charge of the Telemedicine project for Exent. During this project, in 1996, Nave developed remote-control applications for medical systems, which enabled physicians to connect on-line with remote medical centers, thus enhancing their medical resources.
Nave attended the Technion, Israel's Technological Institute, where he earned a Bachelor of Science degree in Computer Engineering.

Dimitris Primpas obtained his Diploma and Master Degree from the Computer Engineering and Informatics Department of Patras University (Greece). He works in the Research Unit 6 of Research Academic Computer Technology Institute (CTI). His research interests include Computer Networks, Telematics, Distributed Systems and Quality of Service.

Alex Shani is the ASP-NG Project Coordinator. Alex has 25 years of experience in the area of telecommunication and networks. Serving as senior manager for Motorola, Alex gained experience in all areas of marketing management, including strategic planning, business development, product development, new product release, and specific country's requirements (standards and regulations). Alex is also consulting in the fields of Telecommunication, Data Systems, Electronic & Mobile Business and Commerce (E/M Commerce), Supervisory Remote Control, Building Automation, both Technologies and Marketing. He serves as a Reviewer and Evaluator for European Commission RTD Program and for the Israeli National Chief Scientist. Alex is Technion (Haifa Israel) BSEE & MSEE (Summa Cum Laude) graduate.

Ohad Sheory is Team Leader at Exent Technologies Ltd. As a team leader at Exent, Ohad Sheory, is responsible for the development of the core technology of Exent's Streaming Application. From 1996 Mr. Sheory managed Exent's Security Section until 1999, at which point he was nominated team leader. Mr. Sheory started his career at Comverse as a Software Engineer for the Digital bit stream recording system. Under that position, he managed to double the system performance, and was responsible for the development of the Built in Tests. He holds a Computer Engineer degree (Bsc) from Technion, Israel Institute of Technology, in Haifa. He received his Bsc in Dean honor in 1994. Between the years of 1986 and 1990, Mr. Sheory served as a Captain in the Israeli Air Force.

Kostas Stamos obtained his Diploma and Master Degree from the Computer Engineering and Informatics Department of Patras University. He has worked for the Networking Technologies Sector of Research Academic Computer Technology Institute (CTI) , Patras, Greece from the end of 1999 until December 2000. Since July 2001 he works with Research Unit 6 of CTI.

Yoav Tzruya is Vice President of Products & Markets Strategy at Exent Technologies Ltd. With over 14 years experience in software development and marketing, Yoav Tzruya is in charge of evaluating potential markets for Exent, defining product strategy and roadmaps, and bringing Exent's messages and products to the market. As a recognized industry expert in the fields of broadband value-added services and business support systems, Tzruya is a frequent speaker at industry leading events. Tzruya came to Exent from TeleKnowledge, a billing & customer care software vendor. During his tenure at TeleKnowledge, Tzruya was responsible for product strategy, marketing, product management and other areas, serving as VP Product Strategy, VP Marketing and VP Product Management. Yoav holds a Bachelor of Science degree in Computer Science and Mathematics (cum laude) and a Master of Science degree in Computer Science (cum laude) from Tel Aviv University.