N-tier E-deposit Infrastructure at University

[1]Ljupco N. Antovski, Marjan K. Gusev, Vangel V. Ajanovski

Abstract - **The e-Deposit is a deposit managed in electronic form. It is appropriate for use at universities. The application is implemented in three-layer architecture. Due to the extensive exchange of financial data, the data integrity is secured. The modular implementation reduces the risk of unwanted behavior. Special modules enable integration of the e-deposit scenario in existing university information system.**

Keywords - **e-Deposit, e-commerce, e-university, PKI, encryption**

I. INTRODUCTION

Deposit account is a demand, time, savings, and passbook or similar account maintained with a bank, savings and loan association, credit union or like organization, other than an account evidenced by a certificate of deposit.

The e-Deposit is a deposit managed in electronic form. The access to the financial funds is granted without physical authentication of the deposit loan owner in the organization that maintains the deposit. The user authenticates himself/herself with security procedures over the Internet.

The e-Deposit is frequently used in the everyday financial transactions. For example, the funds in the e-Deposit are used for distributed money transfer over Internet by e-trading, e-betting or e-auctions [5].

The concept of fund that can be accessed for different financial transaction makes the e-Deposit appropriate for use at universities. The students during the studies pay for different services offered by the university campus.

The question of effectiveness is specially expressed in last minute actions. With deposited prepaid finances, it takes time for a button click to make a financial transaction, instead of bank transactions.

The e-Deposit payment could be used in different areas of student's life. The service offered by this approach implies administrative taxes like exam file, different certificates, semester scholarship, cantina, library, laboratory use and other.

There are no transactions from one account to another, only in the beginning the money are transferred to a specific account. When the deposit is consumed, the student is verified to improve the fund in positive manner in order to be liable for further payments in the university campus.

[1]All the authors are with the Institute of Informatics, Faculty of Natural Sciences and Mathematics, Ss. Cyril and Methodius University, Arhimedova b.b., PO Box 162, 1000 Skopje, Macedonia; Email: marjan@ii.edu.mk, anto@ii.edu.mk, ajan@ii.edu.mk

II. N-LAYER FRAMEWORK ARCHITECTURE

To maximize the functionality and security, the application is implemented in three-tier architecture [1,10]. The three-layer architecture given in Fig.1 consists of the following components: Presentation Layer, Business Layer, and Data Layer.

The presentation layer provides an interface to the end user into the services of the e-Deposit portal. This layer only encapsulates the presentation of the information, but not the business logic [7]. The information received from the business tier is transformed in HTML format and presented to the user on the client's browser. All the demands from the client are sent to the business tier for processing.

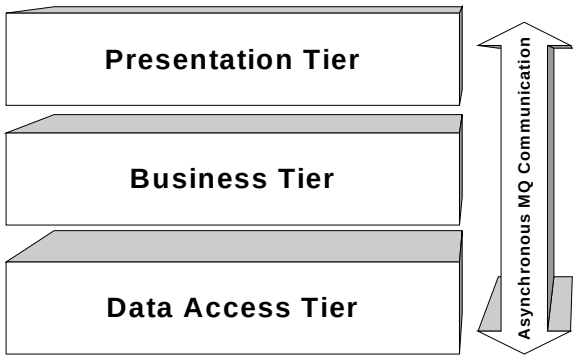

Fig. 1. Three-Layer Architecture

The business logic is processed in the business layer. All the elements of business logic, the rules and calculations are placed in this tier [2]. Every user authentication, transaction demand and verification is executed in this layer. It is a virtual interface among the presentation and the data layer.

The data layer implements a stateless object with generic procedures for connection with the physical database.

Because the processes in the e-Deposit environment are highly asynchronous, considering the distributed client, the business calculus and database communication, a system with asynchronous message queuing is introduced. This system incorporates three public message queues, each for every tier. The tiers communicate to each other in a hierarchical way with short predefined messages.

The messages are XML (Extendable Markup Language) based. The main idea is to transfer information separated from presentation. This type of communication easies the burden on the business and data tiers and enables transformation at the presentation layer not only in HTML format, but also in whatever compatible format with the channel of communication in use.

III. NETWORK IMPLEMENTATION

The network implementation consists of the following items: Client computer, Web Server, Business Server, Database server.

The whole network implementation is given in Fig.2. The network is separated in two parts:

- Front End – the WEB server communicates with the client computer through SSL (Secure Socket Layer) Internet.
- Back End – The Business and Database server protected from outside in private network accessed only by the WEB server in a restricted manner.

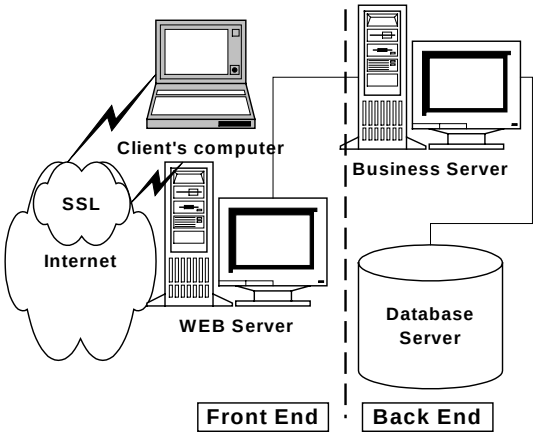

Fig. 2. Network Implementation

The connection among the servers is secured and access is granted to a restricted list of users. The communication is only available between the client and the web server, the web server and the business server, the business server and the data server. The communication is asynchronous and through a system of message queuing. In the final version, the web and the business server are implemented on a single machine.

IV. SECURITY CONSIDERATION

Because there is an extensive exchange of financial data over the Internet, the integrity of the data must be secured. The employed security in the e–deposit solution is: Secure Socket Layer (SSL), Public Key Infrastructure (PKI) and Firewall Security.

The WEB Server uses the SSL protocol to create an encrypted communications channel between the client and server on the transport layer. SSL is a generic "pipeline" that secures data [4].

The Public Key Infrastructure (PKI) is introduced on the application layer. PKI comprehensively satisfies the security requirements of e-Deposit.

A Certification Authority (CA) is a certification-service-provider, which issues public key certificates [9]. The given university comes as a Certificate Authority. It issues certificates and confirms the identity of the distributed users. The certification authority's key is used to sign the certificates and it is identified in the certificate as the issuer. Those certificates are used in the PKI infrastructure for the security procedures.

The PKI implementation covers the following aspect of secure transactions:

- Authentication,
- Confidentiality,
- Integrity,
- Non-Repudiation.

PKI as a compliant security model provides an establishment of a Trust chain, valuable in financial transactions [3,6].

The identity validation is established through various methods of identity check. The methods implemented are:

- User name and password validated on client's side with the use of the login media which encapsulates encrypted user information,
- Cookies,
- Digital Certificates stored on login media (smart cards or mini CD-s).

The installed firewall provides a high level of state-full security between the front-end server and the back-end database and business server. Specific policies are installed only to allow restricted communication.

V. APPLICATION IMPLEMENTATION

The overall e-Deposit application design consists of five independent functional units incorporating most of the application logic. All of these units work concurrently, while some of them running all the time, and some only when triggered by special events.

The five main functional units, bearers of the application logic and design are:

- Deposit management,
- Online shop,
- Request processing system,
- Service broker and
- Delivery system.

The **Deposit management system** takes care of the internal e-Deposit account of the clients (students) and the payment processing. Its work is mainly based on processing two payment transaction queues:

- e-Deposit Incomes and
- e-Deposit Expenses.

The Incomes queue is used for registering bank transactions regarding the transfer of finances from students' bank accounts to their University e-Deposit account. The Expenses queue keeps records of transactions for University Services payments.

The **Online shop** is a web application style offering products for sale. This is done via a classical interface that features notions like "shopping cart" and "checkout". There is a list of products shown over different product categories. The student chooses a product to put into a shopping cart, with the possibility to return the product in case he/she changes the mind. When the student chooses all the products for buying, he/she proceeds to the checkout point. At the checkout, there

is the possibility to authenticate and to buy the products provided the student's e-Deposit account has positive balance.

The cross-functional process style diagram depicted in Fig.3 shows the division of the process activities and the functional units responsible for those activities. Each of the elements (rectangles, etc.) represents some activities to be taken by the corresponding functional unit, and the arrows define the order in which those activities should be done. This flow diagram describes the conceptual level abstraction, instead of the lowest application level.

The main difference this system has over regular online shops is that products offered in the online shop are in fact University Services. Another difference compared to the way regular online shops work is the checkout process. In this application system, during the checkout process, terms of payment and shipping are not specified. This is the e-Deposit concept, since the funds are already transferred inside the institution.

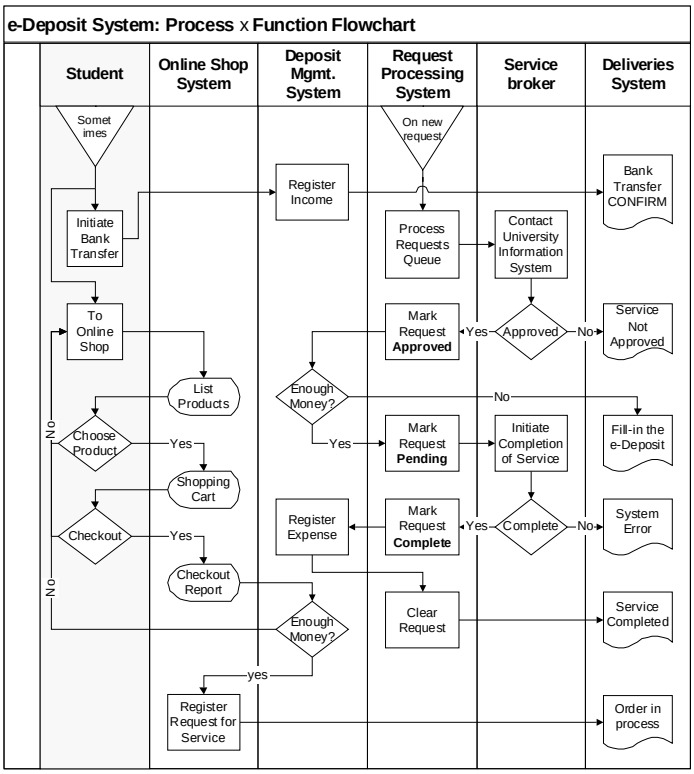

Fig. 3. Process and function diagram

Besides this, the "products" which the money is paid for, are all in fact activities done entirely in the electronic domain for administrative purposes or e-Administration on a way to paperless electronic society.

The result of the student shopping process is a queue of requests for services processed by the **Request processing system**. Successful results are queued towards the Delivery system which informs the user when the service is completed and instructs the Deposit management to deduct the amount spent from the e-Deposit account.

Most Universities are already using some kind of an information system to maintain internal records. Such information systems usually work better if left untouched – so the only solution for collaboration of these two essentially different parts is via some kind of an interfacing system that transfers information back and forth the University information system. The name of this subsystem is **Service broker**, and its main task is to get the operational status and results (reports, forms and documents) from internal components of the University Information System. This is realized in concordance to the service codes and clients' authentication and authorization. Another important task for the Service Broker is the initiation of all the necessary actions that internal functional units should take in order to produce the requested results. An example of operational status is the eligibility of a client for the listed service.

During shopping, the student accumulates a list of requests for service for which he/she is preliminary considered eligible (in a later process this fact is reassured). The Request Processing System processes each record in the list in a sequential manner (one record at a time) and in the exact order, as it was specified inside the online shop, when moving the products to the shopping cart! Failing to process the requests in such way can be dangerous for the integrity and consistence of the data and data structures and could potentially lead the student in an illegal status – depending on the chosen services.

For example, all the students can sign out of the University at any given moment. Most of the students are also eligible to register for different courses and exams. However, if one student makes a request consisting of these two activities in a sequence – register for an exam and sign out, it would be wrong to do them in the reverse way. Approving or denying the requests as a package, or establishing dependencies between them in order to find the right execution order is generally hard to accomplish, and out-of-order execution may lead to unwanted results.

The only real solution for this problem may be the introduction of a special status attribute for all requests. This attribute would state either "Approved", "Pending" or "Completed" in correspondence with the status of each request. The processing of the requests should be done according to the rule – *do* the next request only if all the previous are marked as "Completed".

Once a request is marked "Completed" it means that the service has finished and that the results are sent to the client (whether successful or not successful). The "Completed" status mark also means that the finances required for the service completion are subtracted from the e-Deposit account balance, in the amount specified by the university's regulations. This mark also means that the request records will be cleared out of the Request for service queue once all of the requests in the package have been completed successfully.

After all these necessary steps, the control over the information is transferred to the **Deliveries** subsystem – which acquires the results from the University Information System and hands them over to the student. At this moment, the results are presented as Internet documents. However, the possibility remains open for future expansions towards mobile or other wireless solutions.

VI. DATABASE DESIGN

The described application system structure manifests the need for a dedicated database management system in order to separate this application system structure and the complementary university's information system. This implementation uses a relational database schema that is separate from the university's information system database and in that way highly scalable – as long as the interface to the university's information system is maintained and managed to scale within the same factor.

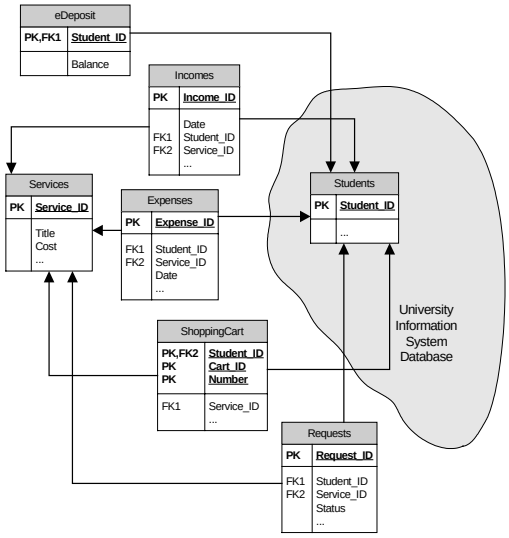

Fig. 4. e-Deposit relational schema

All of the mentioned queues in the application design are implemented as database tables (as shown on the schema diagram on Fig.4) – the list of offered Services, the Expenses and Incomes queue, as well as the Requests for Service queue – all implemented as separate independent database tables. Usage of special fields in the design makes possible the distinction among the different states of activity of the queues.

Management of the possible e-shopping requirements is task that is responsibility of the Online shop system. For this purpose, there is also the Shopping Cart table gathering different combinations of products a client has pre-selected for buying.

The diagram presented on Fig.4 shows a conceptual level abstraction from the perspective of the application system logic, including the key parts of the e-Deposit relational database schema, but it is not a detailed relational database diagram. The diagram on Fig.4 also shows parts of the e-Deposit system interaction with the university's information system database schema.

VII. CONCLUSION

The e-Deposit is used in the everyday financial transactions. The funds in the e-Deposit are used for distributed money transfer over the Internet. This concept proves to be sustainable in the academic environment.

Further more implementing e-university must take advantage of the e-commerce possibilities which are offered in this context. The financial events in the real university life are transferred to e-events as an analogy to the real life where terrestrial trading is substituted with e-commerce.

The main established goal is fast and secure transactions with prepaid service. This service takes an instance of time to transfer financial funds for any reason in the university campus. This concept enables the student to concentrate on the academic, not on the financial side of the university education.

With the procedures implemented in data manipulation, the system is designed to be data bullet proof. The three-tier architecture is upgradeable and scalable. The security measures undertaken make the system unbreakable in a life-time.

This project is extendable to various means of communication, especially to mobile devices. Full integration in the e-commerce environment and enabling deposit payment for different kind of goods and services that are not tightly connected to the university is in further consideration.

REFERENCES

[1] Lj. Antovski, M. Gusev, "Ebanking–Developing Future with Advanced Technologies", *Proc. of 2nd Conf. on Informatics and IT*, 2001 20-23 Dec; Bitola, Macedonia.

[2] T. Edward, *Transactional COM+- Building Scalable Applications*, London, Addison-Wesley; 2000.

[3] C. Elison, B. Schneier, "Ten Risks of PKI: What You Are Not Being Told about Public Key Infrastructure", *Computer Security Journal*, Vol.16, N.1, pp.1-7.

[4] A. Freier, P. Karlton, "The SSL 3.0 Protocol", *Specification*, Netscape Comm. Corp; 2001.

[5] M. Gusev, "E-Commerce, a big step towards E-Business", *Proc. of 2nd SEETI Conf. on Trade Initiative and Commerce*, 2000 Nov. 8; Skopje, Macedonia.

[6] M. Howard, *Designing Secure Web-Based Applications for Windows 2000*, Microsoft Press, Washington.

[7] Microsoft & Cisco, *E-Commerce Framework Architecture Document*, White Paper; 2000 Nov.

[8] A. Milosevic, "Bank Information System Architecture-Integration in Existing IS", Mircetic M, editor. *Proc. of New Banking Vision Conf; Ohrid,* 2001 Feb 21-23; Pexim, Skopje; pp. 61-67.

[9] R. Rivest, A.Shamir, L.Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, Feb.1978 Vol.21, pp.120-126.

[10] K.Terplan, OSS Essentials-Support System Solutions for Service Providers, 2001 Feb, Wiley, Canada.