REALIZATION OF PAYMENT SYSTEMS BY STANDARD APPLICATION COMPONENTS

Slobodan Babic, Ivan Bojicic, Ivan Tamburic
SAGA d.o.o. Belgrade, eBusiness Department, www.saga.co.yu

ABSTRACT

The realization of payment systems is a very demanding endeavor due to multidisciplinary elements having to be embedded under very strict criteria arising from the very nature of financial systems. The SAGA eBusiness Department entered into the payment system field at the moment when Serbia was going through the preparatory stage for payment system transformation. Since then, high requirements have been imposed for payment systems which has directly influenced the continual efforts in creating adequate payment system structure. The results of such efforts are presented in this paper in which the essential system elements are described, as well as the positive results of system element standardization on all levels of system realization. The paper also describes the realized system for check clearing, implemented at the Association of Serbian Banks (ASB) which was awarded with the prestigious award of the Banking Technology Magazine: "The European Banking Technology Award 2005" as the best solution in the check clearing and payment solutions group.

1. INTRODUCTION

By introducing a new payment system on January 6, 2003, the National Bank of Serbia (NBS) created favorable conditions for consistent development and regular overview of financial systems operating issues. In that way the NBS provided possibilities for software companies to develop information systems of the highest level in global terms. This is evident in the system standardization possibility. System standardization tasks refer primarily to the identification of common properties of business processes and their unification. That way the IT support is simplified, the implementation and maintenance are made easer, and the IT support gets all other affirmative attributes serving the technology - operating process. The standardization tasks better determine the technology environment supported by the IT technology.

SAGA eBusiness Department (SAGA eBD), has for several years been developing the RTGS, ACH, and related systems based on SWIFT [4] and other standard messaging systems [5]. The development experience is used for the suitable choice of technologies which fully satisfy the technical, functional, and non-functional requirements, and industrial standards which could be posed to such a product by a potential market.

Our view of payment systems is presented in the following chapter and it unites all relevant elements present in every payment system, namely: technology elements of financial institutions implied by the messaging standard, environment conditions posed by the location in which the payment system is realized (the place in financial hierarchy, the type of processing preferred, the geo-political topology, and other potential elements), required communication interaction, necessary information technologies for the realization of particular and general tasks.

2. PAYMENT SYSTEM STRUCTURE

The financial message format is such that it contains the information necessary for the very message transmission from one to another location, as well as technology information – data on the business process they are intended for. They provide the possibility for determining the logical object-oriented system where the objects are in interaction solely by means of the messages. One such a messaging system is the SWIFT messaging standard which in itself encompasses combined elements (for transport and processing). The other financial industry messaging standards (from phenomenological standpoint, this is to be expected) also contain the information necessary for transport and processing (some of them are highlighted in Figure 1.).

Selecting the appropriate sets of messages of the adequate standard we are able to "cover" a financial part. The chosen subset of standards defines the appropriate logical whole. Check clearing system is a "logical part" based on almost identical messages as "Credit transfer" but having logical limitation of being valid only for checks, and thus presenting a separate logical part. Therefore the logical parts are determined by the process owner norms but in accordance with permissions/approvals/contracts which he has in his environment (surrounding) relative to other participants.

The physical systems are actually the consequence of process owner organization and they are in organizational terms dedicated to, for example, Payment house, if the owner has the right to found such an organization, or some others, but it is obvious that some of them are dedicated according to financial or geopolitical hierarchy.

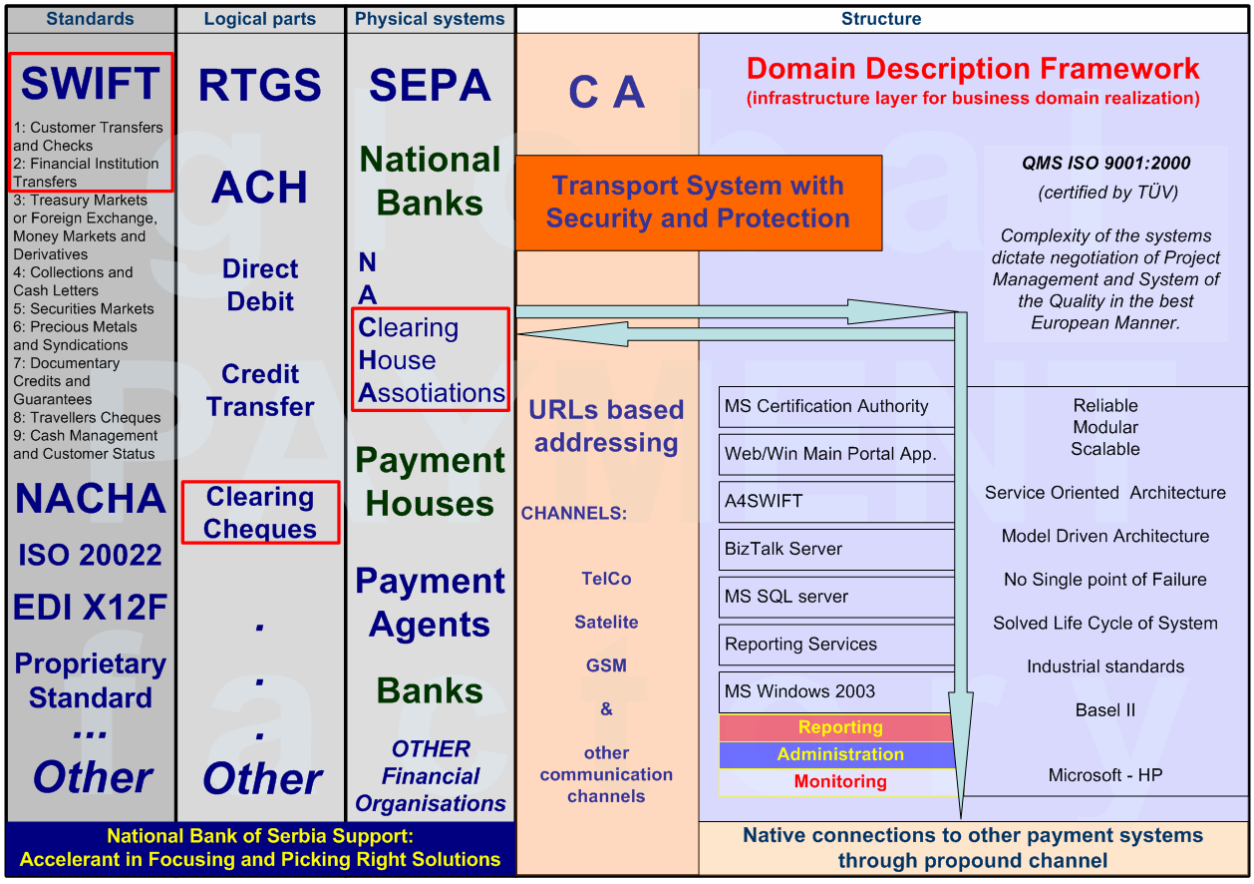

Figure 1. Realization System Selection Table

3. STANDARD COMPONENT PRESENTATION

The set of standard components was developed with the aim to speed up and standardize the realization of complex information systems. All standard components are defined so that they can support a large number of different systems and are thus reused in development of new information systems.

Transport system is very flexible for the new requirements posed to such a system by the “eccentric” financial market and contains an advanced security and protection system. For system connection, a minimal requirement needs to be met, the addressability of the URL nodes, so that it is transparent relative to all other elements which are possible to be defined, at the same time being reliable and safe. It can be realized on TelCo resources, using mobile telephony, by satellite communication or combined – even depending on current conditions present in the system. It is possible connect the system to other message transport systems in an arbitrarily-defined reasonable way through a dedicated communication channel (SWIFT, NACHA, SEPA, and other messaging exchange systems).

There are several messaging exchange systems in the world but they are all unified through SWIFT network dispersion. The messaging exchange systems contain different standards which not so seldom present the basis for the monopoly of network owners towards the participants – regardless of the way the network owner presents himself. The network ownership is then unnatural and the participants are forced to push their contents through these channels. The unnatural feature is reflected in the possibility of content monitoring during transport, even the control (we assume that there is no monitoring and control but that such possibility exists). This resource is strategic for each country individually, as well as for states grouped around some interests, and SAGA eBD Transport System contains the required solution.

Process Workflow Description Framework is the component which enables formal description and automation of business processes. It consists of two basic parts: custom developed *Workflow* system, based on Workflow Management Coalition standard, responsible for business process description, and *BizTalk Server*, based on XLang language for description and automation of business processes, through which the activities of defined processes are described. This component can describe a wide spectrum of various business processes and it is used for formal description and behavior automation of the relevant system.

Domain Description Framework [1] (application framework) is custom implemented system of median layer components. Domain Description Framework unites system functional subsystems while administration, monitoring and reporting are possible on the level of the system components, as well as on the level of technology process. The application framework is a set of components realized in the Microsoft SQL Server/C# environment with the aim to provide a general infrastructure layer for the realization of the concepts of a domain. As such, the framework is general purpose software, which is extendable and reusable. The application framework has appropriate general software design rules and principles embedded, inside which the new rules (or specific solutions) are developed (or derived) specific to the appropriate problem. The embedded rules usually consist of one or more design patterns thus in advance defining the software architecture and future application design.

4. LOGICAL ARCHITECTURE OF A CLEARING HOUSE

The system is based on service-oriented architecture. The logical system architecture presents a stable basis for system development and enables the system to be realized as flexible enough to successfully respond to functional, organizational, technological and other changes in the environment and the system itself. In such a way it facilitates not only the automation of business processes but also their improvement.

As shown in Figure 2, the system is modular and mapping to the adequate physical nodes provides performance enhancement by simple adding and/or replacement of hardware components. The system can be divided into the following functional components:

1. Business logic components
    1.1. Process Description Framework
    1.2. Domain Description Framework
2. Portal, Monitoring and Administration
3. Reporting
4. Relational System for Data Base Management
5. Transport system

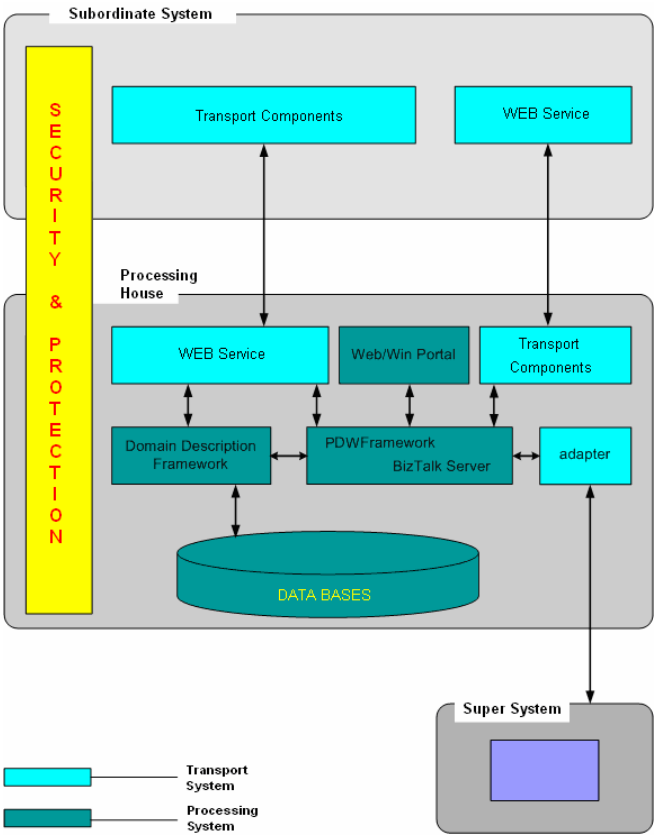

Figure 2. Logical System Architecture

In development planning and system architecture creation, the starting point was the basic requirements posed to the project, which is the complete fulfillment of functional requirements, as well as very high availability, high performance, reliability and scalability. The system is envisaged as highly accessible, implying the operational system functionality which is very close to or equal to 100% up time. Regarding the system performance, the requirement is set for processing of minimum 50,000 inbound and 50,000 outbound messages per hour. Regarding the scalability the requirement set was that the increase of capacity/system performance can be easily achieved by multiplication of existing system parts (e.g. by adding new servers).

5. CHECK CLEARING SYSTEM REALIZATION

The foreseen methodology of system development is a modified unique process of software development, which in its basis tracks the agile iterative-incremental life cycles of development [2]. The methodology defined in this way is combined with current achievements in the field of complex information systems development, primarily in the domain of multilayer architecture based on distributed components and services. The realization of check clearing system was taking place in two parallel stages: (1) specification of conceptual solution and (2) realization of standard components. In a given point in time these two stages merged, by which the conclusions from the design stage were applied on standard software components. All the components of the resulting physical architecture of Check Clearing System are presented in the following paragraphs.

Process Workflow Description Framework presents the component which defines system functions (system behavior) and is realized so that it enables the formal description of business processes. One of the requirements for the development team related to the possible change of business process description, as well as to the existence of conditions which can influence the business process during execution. In order to make the business process flexible, our own workflow system was developed describing the entire process, while the BizTalk orchestrations are used for the observed-process activity execution.

Domain Description Framework is a component which defines system structure, and it is realized through the following components: IS Dictionary – (storage of models and rules of transformation), Predefined Software Architecture – (design of future applications) and Code Generator – (automated implementation of transformations). The IS Dictionary [3] describes the whole system, that is all identified business and technical domains. At that, the IS

Dictionary defines the set of concepts by which the appropriate system is formally described, while these concepts are sufficiently general to be able to describe a large number of different systems or their parts. *Predefined Software Architecture* is a set of realized software components which embed rules for design by which the architecture and design of future applications is defined. In addition to the design, the infrastructure layer is provided which can support a wide spectrum of systems and containing parts for transaction control, materialization/dematerialization and object cashing, and keeping record of users accessing the system etc. Code Generator is the component responsible for automatic model transformation into an executable system. The transformations are based on the description of technology platforms and formal defining of transformations. This facilitates for the model which describes a business domain to be automatically transformed to the appropriate target platform.

Portal, Offline Application, Monitoring, and Administration are the applications facilitating easy interaction with the system. The portal application provides the adequate reception, i.e. expediting of documents and their recording, as well as the necessary procedures for the execution of given operations on documents. Offline Application facilitates adequate reception, i.e. document expediting and their recording (similar to portal application) but with the all its functionalities determined relative to the communication channel which is open only from time to time. The real time application for monitoring enables the real-time presentation of the given data sets on system status which are of importance to the user at that moment. Through transparency and easy selection of details (e.g. by mouse click on a particular area), Monitoring also provides the overview of appropriate information which are hierarchically below the details. System administrator can implement requirements arising from the functional or application reasons by system modification through the administration system.

Reporting is implemented using the Microsoft SQL Server Reporting Service. The reporting services allow the centralized management, easy expandability, scalability as well as high data security, interactive, proactive, and passive reports in many various output formats such as HTML, EXCEL, WEB ARCHIVE, ACROBAT (PDF) FILES, TIFF, CSV, XML file with report data. They have embedded support for report generating from a large number of data sources. All reports as well as resource files mounted on the report server are organized in a logical folder hierarchy above which there is and extendable, role-based security system integrated with computer/domain users and groups.

Database is a part of the system which has to provide high accessibility, reliability, and high performance. The accessibility refers to the time percentage in which the system is accessible to the user. It is achieved by the system down protection, providing system redundancy in case of system down, and by system segmentation. SQL Server 2000 and Windows 2003 offer several solutions for meeting these requirements: Failover Clustering, Database Mirroring, Log Shipping, Data Replication, SQL Server Federations.

Transport system is realized as a proprietary network for message exchange**.** There is no limit in terms of contents, namely, other contents can also be defined, including the transmission of binary files between participants. When elaborating on this system, the system characteristics should be taken into consideration, such as speed of implementation, ease of application and maintenance, scalability, security infrastructure based on industry standards, “cross-platform” support, and satisfactory performance, as well as general service-oriented system architecture, which depends exclusively on system installation requirements.

6. PHYSICAL SYSTEM ARCHITECTURE

The system is realized by Hewlett-Packard, APC, and Cisco components. Based on calculation and estimation, the hardware structure is defined to satisfy the system requirements completely and optimally. The proposed system consists of three subsystems: Production System, Back-up System, and Test System, which fully meets strict conditions of financial industry systems. The system is constructed to avoid “single point of failure” on the primary and back-up location. The role of the test subsystem is to check the functionality of system components during the life cycle of the production system. One of possible realizations of the Production System is shown in Figure 3. In addition to this one, there are also other possibilities, depending on owner business policy and prescribed system standards.

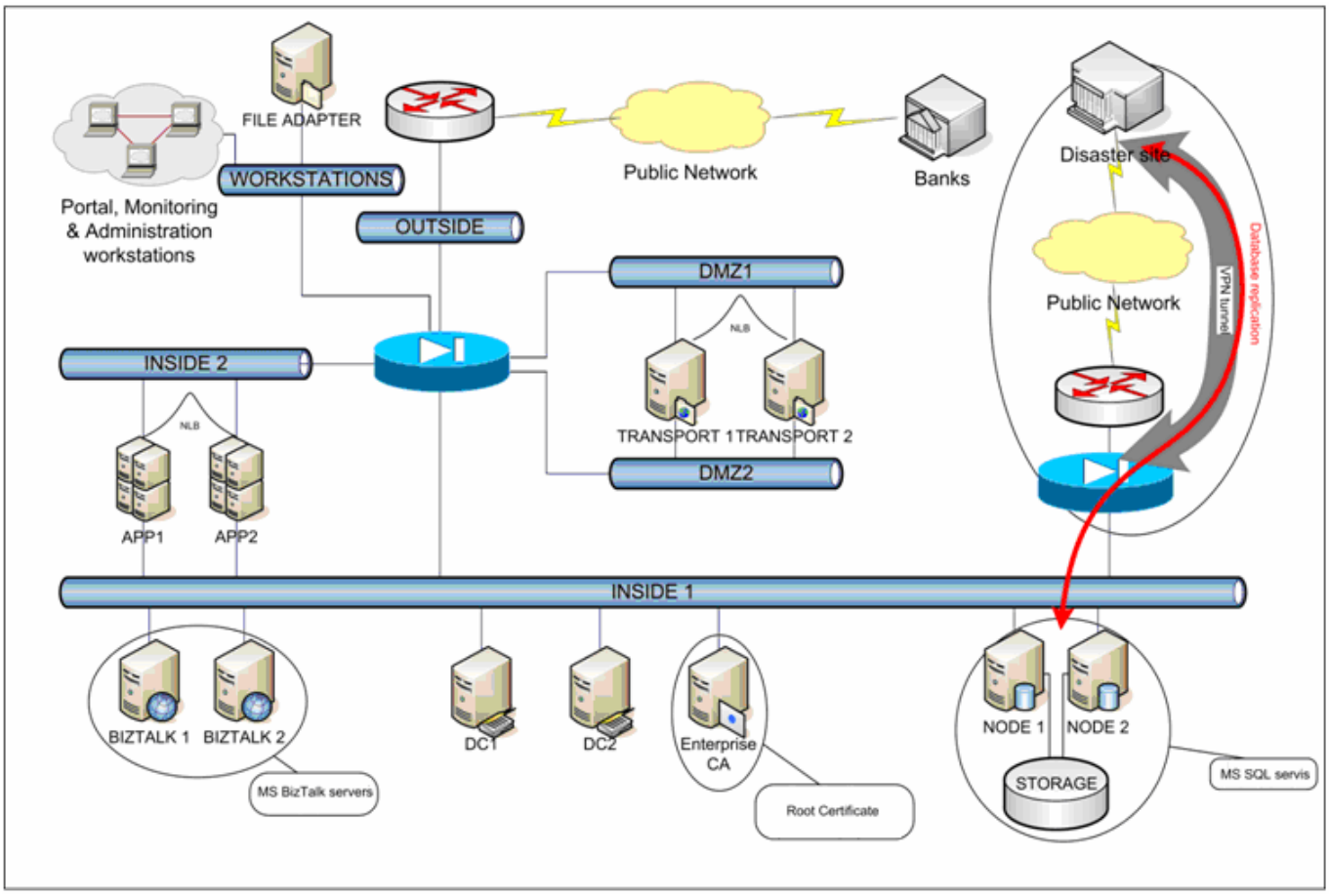

Figure 3. Potential structure of the Production System and inter-dependency of its hardware components

7. CONCLUSION

Hereby also we wish to express our satisfaction with the collaboration with the National Bank of Serbia, Serbian banks, and the Association of Serbian Banks (ASB), as we have realized the solution for check clearing with their participation and assistance. The achievements of this solution are:

- The Serbian banks and other financial organizations – the participants in process of check clearing, can communicate with the full set of SWIFT messages.
- The installed system enables the ASB to accept the message processing for any financial system (regardless of geographic location) based on the set of SWIFT messages without significant investment in the system, by adding new application modules to the existing system.

The system production started on May 9, 2005. In the first year of exploitation there was “zero downtime”, additionally confirming the flawless functioning of the system. In support to the above stated, we would like to mention that for the mentioned system the ASB was awarded with a prestigious award of the Banking Technology Magazine, “The European Banking Technology Award (2005)” for the best solution in the clearing and payment solutions group.

8. REFERENCES

[1] Bojicic I. *Domain Description Framework – Software Tool for Support of Model Driven Software Development* in InfoM Journal for Information Technologies and Multimedia Systems, 12/2005.
[2] Jorn Bettin. *Model Driven Software Development* in MDA Journal, April 2004. Available at: http://www.bptrends.com/publicationfiles
[3] Bojicic I., Lazarevic B. *Aspect Oriented Approach in developing IS Dictionary* at Yu Info, Kopaonik, March 2004.
[4] *Society for Worldwide Interbank Financial Telecommunication* (SWIFT), www.swift.com
[5] *Transaction Workflow Innovation Standards Team* (TWIST), www.twiststandards.org