Realization of Transport System in financial industry

***Radoslav Petković**, SAGA d.o.o. Belgrade, Electronic Business Department*
***Slobodan Babić**, SAGA d.o.o. Belgrade, Electronic Business Department*

Abstract
The systems in financial industry belong to the "mission critical" system group. Because of that they should satisfy very high criteria and at the same time to offer the maximum comfort in usage. The starting systems premises in financial industry are given by law regulative, juridical acts and normative which determine the electronic documents – message systems and business processes which have to be done on these documents. The crucial processes in financial systems are connected for electronic interchange of financial documents. In the article SAGA SEP Transport system v3.0 is presented for electronic interchange of financial messages which besides the stated features satisfy the set of other ones and makes it the basic system for every financial system nevertheless to the relative position – the system level in the vertical financial industry hierarchy.

1. Introduction

Financial systems are by their nature qualified systems with clearly separated authorizations and proposed responsibilities. They are qualified because they are intended to do specific type of job or a spectrum of jobs similar by their nature. They are authorized by the appropriate institution (by the Law of Central bank, banks and other financial institutions, and other government regulations) and Central bank [3] (juridical records and normative regulations) by which is defined structure of responsibility.

There are two groups of records in financial systems: data to be processed and data about processing. The entities of financial systems mutually exchange the processing data by the given business rules and with appropriate database on the location of receiving.

The processing data are stored in electronic document which must satisfy the following criteria in order to belong to the system:

1. Electronic documents are to be exchanged by the agreed communication channels in the form of formatted data.

2. Electronic documents are grouped by criteria of belonging to the same set of data needed to be realized business process or a part of business process.

3. Electronic documents are to be unmodified on the address of the receiver with only allowing adding of data to the initial set in order to fulfill adequate law regulation.

4. Electronic documents are to be processed in the real time (real time is determined only by the business processes from which follow data to be exchanged, so the basic unit of measurement is not common – response time in the system).

5. Electronic documents are not to be such to allow systematic listening in communication channels and such that there is not possible systematic acquisition of data.

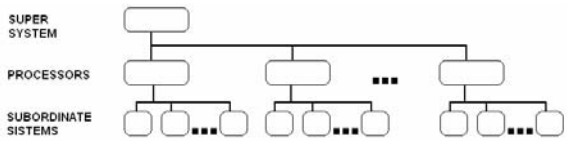
6. Electronic document on the receiving address must be non-repudiated.

7. Every sent message (electronic document) must have its acknowledgement. Receiving acknowledgement is also a message (electronic document), that satisfies all the criteria of message but doesn't have message acknowledgement.

8. In the case of expertise of the authorized institution, with help of the distribution set of messages and replies, there must be possible reconstruction of events flow and business process that is expertised.

Every financial system consists of the following components (Picture 1.):

1. Super system – superior center,
2. Processing systems controlled by super system
3. Network structures of subordinate systems – participants in „traffic".



***Picture 1.** Relations between systems*

Every participant has unique instances of electronic documents that have participated or will participate in business processes. Electronic documents are owned by super system (managing center data and data about processing in the processor system), processor system (processor system data and data about subordinate system message processing) and subordinate system data. Listed groups of data make unique entity, they must be kept determined period of time, and in the case of failure, at any level, of any electronic document (or set of electronic documents) results with serious damage of financial system elements that are balanced by juridical and punitive regulation (missing of electronic documents or their destruction in most cases means serious criminal act).

1.1. Message Oriented Communication

1.1.1. Data object (DO), in paradigm of object oriented programming, is individual executive unit of current data used as a basic building programming block. DO influence mutually, so compared to the traditional view, DO may be interpreted as a collection of functions or simply a list of computer instructions. Every DO is capable of receiving messages, processing data and sending messages to the other objects. Every DO can be seen as independent little machine or participant with strictly defined roles and responsibilities. In our system DO is Message Object, message which transmits informative content between two systems.

DO extends usual (single-process) models of programming into distributed environment. DO made communication transparent or implicit, introduced new distributed application programming model based on explicit communication, which is, in the case of SAGA SEP Transport v3.0, persistent.

1.1.2. Persistent communication assume that receiver and sender needn't be connected at the same time. Communication channel in that case has power to preserve message until the remote system is connected.

1.1.3. Message is, generally speaking, object for communication. It provides information, and can be information by itself. Its meaning depends on context in which appears – is used. The term "message" can be applied as well to the information as to form.

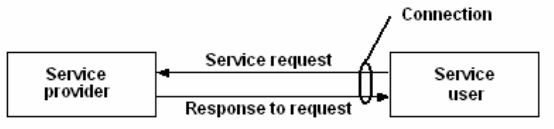
1.2. Windows Communication Foundation (WCF)

WCF is Microsoft development framework for developing secure, reliable, transactional and interoperable distributed applications.

WCF incorporated programming model that uses managed code for developing unique WEB services and other distributed systems which can communicate mutually. WCF focuses on connecting XML with programs developed under Microsoft supported development languages VB.NET and C#. To support this interlingua communication, WCF uses Extensible Simple Object Access Protocol (SOAP).

1.3. Service Oriented Architecture

Service oriented architecture is basically set of services which communicate to each other and can include simple data exchange but also include two or more connected services to execute some activity. Picture 2. shows basic concept of service oriented architecture. Service consumer sends request to the service provider. On that request, service provider returns adequate response to the service consumer. Request and appropriate response are realized over the connection which is determined and which both sides can understand. It's interesting that service provider can also be service consumer.



Picture 2. The basis of service oriented architecture

2. Transport system description

SAGA SEP Transport system is projected to provide safe, reliable and efficient message exchange, based on SWIFT [1] and other industrial standards [2] between participant in system. System can be replacement for traffic over SWIFT network. It is cheaper and self owned. First version of this system is implemented in Association of Serbian Banks - Clearing Institution of Banks (ASB - CIB).

2.1. Project requirements

System is modular, available, stable, scalable, with appropriate security mechanisms and high performance. It is based on Microsoft technologies so it minimize operative risk of implementation technologies (according to BASEL II) because all other systems in that class are realized in different technologies.

The system is also required to have characteristics to be widely used in the context of transparent usage independent of geopolitical or any other configuration of participants, not limited only to financial industry domains, and with possibilities to include similar processes supported with accepted standard for message specification, and to support all other systems based on communication and exchanging of messages which consists embedded information about receiver's address and business process needed to be done at receiver's location. Table 1. shows standard project requirements for such systems and description how each requirement is satisfied in the solution implemented in ASB - CIB.

Initial requirements defined for the first version of transport component are copied to new version. Differences are mainly in used technologies. New Microsoft technologies enable wide spectrum of defining communication channels, type of processing, adding various programming libraries, possibilities of defining architecture on highest theoretical standards, implementing wide spectrum of configuration parameters for system determine and achieving strong feedback with possible modifiable system environment, freely defining communication with required external systems throw service interchange,

defining any workflow mechanisms and with possibilities of reaction on events, independently the event came from external or internal source.

No.	Requirement	Description of satisfied requirement
1.	Availability	24 hours x 7 days in week
2.	Safety	Security based on using public key certificates (Microsoft Enterprise CA with public key infrastructure - PKI)
3.	Performance	Tested on 500k transactions per hour (frame relay 512k)
4.	Stability	Version 1.0 has "zero down time" in the first year
5.	Modularity	Can be added modules (global and special) in the case of growing functional requirements, e.g. for delivering messages to direct debit processing system, forced payment, clearing or any other processing system.
6.	Scalability	In the case of system growing, it can be added appropriate number of hardware components

Table 1: Basic project requirements and their satisfaction

2.2. Functionality

Supported functionalities are the following:

- Transport of SWIFT messages and other defined standard documents,
- Message interchange by the folder systems, MSMQ or appropriate application adapters,
- Validation of messages (first three blocks of SWIFT message),
- Digitally signing of every message and verification on destination,
- Five modes of transport (synchronous - asynchronous combination with sender and receiver and message acknowledgement– ACK / NAK), and fire & forgot mechanism (streaming mode),
- Message routing throw defined routes,
- Message archiving,
- Existing of application components for monitoring sending and receiving of messages.

Described functionalities of system are highlighted as the most characteristic and the most interesting regarding to technologists of financial industry.

2.3. Integration

System is independent of the communication infrastructure. There are cases from real word (caused by elementary disasters) where communication is made with mobile equipments without additional notes to other participants. The only requirement for communication is that all participants are URL addressable. It is possible to integrate this system in almost all well known systems for data exchange.

2.4. Security

Security is based on PKI. It gives guarantee that every message will came unmodified and with proof of authenticity. Here is an option for channel encoding (encrypting). PKI is based on Microsoft CA architecture.

2.5. Performance

Described concept is verified in Microsoft Technology Center (EMEA) in Copenhagen, Denmark in the march 2005.

More sophisticated testing is arranged in Microsoft Technology center and have intention to proof that Transport system v3.0 is capable to satisfy requirements for document transfer even for biggest countries like EU, China, India or USA.

3. System architecture

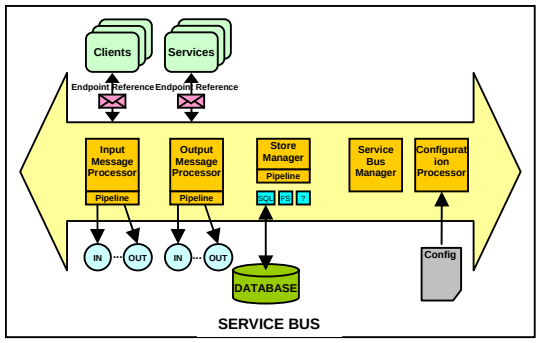
System is based on WS-* architecture [9]. Basic feature of this architecture is composeability. This architecture enable incremental development of Web services throw single requirements (security, availability, stability, packet attaching, filtering ...). Every of this requirements is solving independently of each other, resulting in simplest development.

3.1. Message Bus

The implementation realizes BUS architecture [10]. SERVICE BUS represents logical connection of mutually connected systems. At the same time it enables separating levels of business logic from communication level. Basic features of system are:

- Full configurability (Configuration processor)
- Services based on WCF technology
- Service loader and manager
- Initial set of message processors (Message sender, receiver...)
- Embedded auto archiving system
- Store manager
- Configurable service preprocessors, postprocessors and filters
- Configurable service adapters
- Simplest grow of functionalities (adding new services)
- Embedded security mechanisms.

SERVICE BUS architecture is shown on picture 3.



Picture 3. *eMessaging Transport System Service BUS*

3.2. Managing processors

3.2.1. Configuration Manager

Configuration Manager provides easy and simple system configuration:

- Adding, deleting and configuring WCF services,
- Adding, deleting and configuring filters, pre and post processors,
- Adding, deleting and configuring service adapters,
- Defining and changing application configuration parameters of system elements.

System configuration can be stored in configuration files or any other media supported by Store Manager (e.g. SQL database).

3.2.2. Loader (Service Bus Manager)

Purpose of loader component is to start and stop services according to defined configuration parameters. Every service works in his own application pool. Administrative functions of Service Bus Manager enable on-line communication with services, monitoring of workings parameters, performances and adjusting of some working parameters with or without stopping of relevant services.

3.3. Message processors

Message processors are plug-in WCF services. Every Message processor can have one or more preprocessors, postprocessors and/or service adapters. This is defined by configuration (Loosely coupled). Every main message processor in the system is called Action processor. Action processor can communicate with other parts of system via attached configurable adapters.

3.3.1. Preprocessors and postprocessors

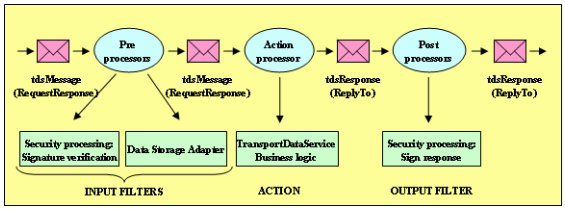
Message processors in Transport system initially have pre and postprocessors that implements security, message integrity, sender/receiver authentication and message archiving. Filters are also preprocessors or postprocessors but needn't to know internal structure of message and they are used for processing at global level (e.g. traffic monitoring, statistics or logging).

3.3.2. Adapters

Adapters are configurable programming elements which provide one or more specific functions, according to system configuration. Initially there exists an embedded adapter for message storing in folders and into SQL database and for communication with BizTalk server.

3.3.3. Input Message Processor

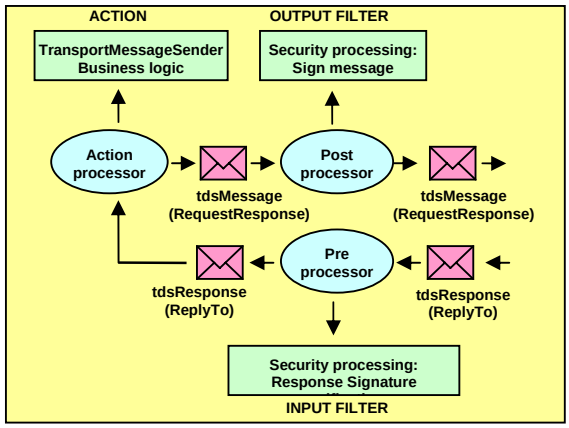
Input message processor in message confirmation mode is shown on picture 4. Message confirmation mode means that the service sends reply message (acknowledgement) back to client. In the case of Fire & Forget mode there are no postprocessors and output filters.



Picture 4. *Transport Message Receiver Service, Message confirmation mode*

3.3.4. Output Message Processor

Output message processor in Message confirmation mode is shown on picture 5. Message confirmation mode means that the processor expects reply message (message acknowledgement). In the case of Fire & Forget mode there are no preprocessors and input filters.



Picture 5. *Transport Message Sender Service, Message confirmation mode*

3.3.5. Message formats

Every message consists of message body and one or more message headers. Message body is technological message in predefined format (e.g. SWIFT). Message headers can be defined by system and by user. In transport system there are defined two types of messages: active messages (transfer) and reply messages. Active message has the following headers:

1. **Action** (system: function name in receiving service),
2. **MessageID** (system: SOAP message identifier),
3. **To** (system: destination service address),
4. **ReplyTo** (system: service address to receive message acknowledgement, frequently sender address),
5. **MessageFormat** (user: message format – swift, text, xml, binary ...),
6. **MessageIdentifier** (user: user message identifier - reference),
7. **MessageSubtype** (user: message subtype, e.g. swift type MT101, MT102, MT196 …),
8. **MessageOrigin** (user: sender logical address),
9. **MessageDestination** (user: receiver logical address),
10. **DigitalSignature** (user: qualified digital signature on message - XML format).

Reply messages have the following headers:

1. **Action** (system: function name in service expecting acknowledgement),
2. **RelatesTo** (system: related SOAP message identifier),
3. **MessageIdentifier** (user: related user message identifier - reference),
4. **ResponseType** (user: reply message type, e.g. ACK or NAK),
5. **DigitalSignature** (user: qualified digital signature on response - XML format).

3.3.6. Receiving modes

Receiving message service has the following modes of accepting messages:

- Synchronous
- Asynchronous
- Fire & Forget (no answer).

3.3.7. Physical channels, protocols

Physical channels / protocols for message transfer is defined by the configuration, message processors are absolutely independent of communication protocols. System is tested on the following transport protocols:

- HTTP
- HTTPS (SSL)
- TCP/IP
- MSMQ
- Message PIPE

3.3.8. Message dispatcher

Every swift message has logical destination address, concrete swift code of institution to whom the message is sending. Message dispatcher uses this code to get destination URL address (mapping is done throw configuration manager).

3.3.9. Security

System consistently applies WCF security mechanisms:

- Message layer security for message authentication
- Transport layer security for confidentiality
- Using of X509v3 certificates as security tokens
- Direct / brokered authentication.

On the application level, there is applied additional protection e.g.:

- Encrypting of message contents
- Mutually end user authentication.

4. Transport System Extensions - New services

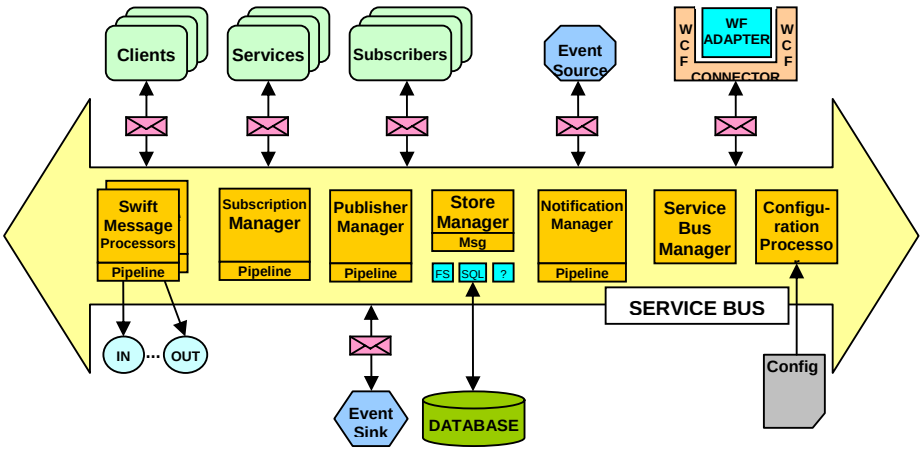
The whole system is based on WS-* specifications and bus service architecture and enables very simple adding of new functionalities by configuring new service that implements this functionality. As a sample we can show some subscriber services as subscribing on foreign currencies exchange list or weather forecast.

To enable those functionalities we add the following services:

- Subscriber service (Subscriber)

- Notification Manager service
- Publisher service
- Event source (as usual, out of this system - remote client)

Extended Transport System is shown on picture 6.



Picture 6. *Extended Service Bus Architecture*

5. Usage of Transport System

5.1. Clearing

Banks using processor's transport system sends messages until defined time and after clearing processing gives back, over same channel, adequate messages about results of clearing process.

5.2. STP

Corporation A pays corporation B by message which is produced by their self and sends it to bank A over bank's Transport system. Bank A, unchanged message, sends it to processor which is routing this message to bank B which sends it to corporation B using transport system.

5.4. eBanking

User of eBanking system forms transaction, sign it and send it over Transport system to bank's core system which process that transaction. Bank's reply is done via the same channel.

5.5 Core banking

Bank entity sends instruction of responsible person in message over Transport system to other bank entity. Reply of other entity (responsible person) is done via the same channel.

6. Conclusion

SAGA SEP new Transport system is successor of Transport system installed in ABS – CIB. The previous system gave considerable results and performances. New system enables connection of financial processes horizontally – in the environment of similar institutions or in one financial institution vertically between financial institutions of different rang. Communication is defined by normative acts of relevant institutions; system supports industrial standards, both in financial area and information technologies. Following that, Transport system can be used as logical base above is possible renewing of existing system of financial institutions that will provide all benefits of new technologies, but also will enable legacy systems to functioning and growing naturally with keeping their investments. This Transport system gives possibility of using distributed services, so keeps possibility of implementing innovations in the area of financial industry and is compatible with all new trends, regulations and initiatives of European Unity [4][5][6].

7. Bibliography

[1] Society for Worldwide Interbank Financial Telecommunication (SWIFT), www.swift.com
[2] Transaction Workflow Innovation Standards Team (TWIST), www.twiststandards.org
[3] Narodna banka Srbije, www.nbs.yu
[4] European Central Bank, www.ecb.int
[5] European Commission, www.ec.europa.eu
[6] European Payments Council, www.europeanpaymentscouncil.org
[7] ISO20022, www.iso20022.org
[8] ISO15022, www.iso15022.org
[9] WS-Transfer, WS-Eventing, and WS-Enumeration Specifications Submitted to W3C.