Analysis of Platform Dependencies in Software Solution for Auction and Trading in Electric Energy Market

Milos Gajic[1], Marko Djukovic[2], Sasa Devic[3], Branislav Atlagic[4], Zvonko Gorecan[5] and Dragan Tomic[6]

Abstract – **This paper shows possibilities and limitations of mobile technologies (*Android and Windows Mobile*) as well as web solution (*Silverlight*) in the management process of auction trading of power energy in Central - Eastern Europe. Also, there is a brief overview of the process of auction and the information that is of interest to the participants. Moreover, it is shown a short description of these technologies and the architecture of the proposed solutions and their performance.**

Keywords – **Android, Windows Mobile, Silverlight, Auction, Energy Market.**

I. INTRODUCTION

Electric power industry is dealing with problems of generation, transmission, distribution and consumption of electricity. The main aim of its activities is to ensure the required supply of electricity to consumers, the prescribed quality, the necessary levels of safeness, reliability and with the minimum costs. Cessation of large investments in the construction and development of electric power system, it leads to the introduction of deregulation in the vertically-integrated, monopolistic oriented electric power companies and creating the need for an open electricity market in the form of stock exchange. Electricity trade in Europe is done through the auction office established by the association of electric-power companies. The auction office should ensure a smooth exchange of electricity between producers and consumers on a daily, weekly, monthly and yearly basis.

The auction process consists of two steps. The first step is trading of electrical power between producers and consumers, while the second step is validation of transactions taking into account constraints of transmission network. For validation of transactions is developed Power System Analyzer (*PSA*) software support. TSO (*Transmission System Operators*) has enabled communication with PSA through a desktop client called PSAClient. Based on a model of the national network that provides transmission operator, PSA performs the allocation of transmission capacity of interconnecting lines at the Central and Eastern Europe, which includes the following countries: Poland, Czech Republic, Slovakia, Hungary, Austria, Slovenia and a part of Germany.

The auction process is very dynamic and often happens that a certain group of people needs access to current network status, as well as to the results of calculation. Still, due to the nature of their work, they do not have a constant access to the auction process information, which could be a problem. There is a probability that having only a desktop client might be a problem. In today's world of information technology it's almost mandatory to have a browser and mobile access for any network application in order to gain a head start over competition.

In order to overcome these problems it is suggested two mobile solutions based on Windows Mobile (*WM*) and Android platforms and one web solution based on Silverlight technology. Mobile phones allows flexibility, mobility and constant access to auction process, but if user needs detail information it can access them via web application suggested in this paper. All solutions, will be presented here, are designed to work alongside with existing *PSAClient*.

II. ARCHITECTURE OF PSA SOFTWARE SOLUTION

Software applications described in this paper are developed to control auction process for trading of electricity. Each country (the operator) of auction process must have an insight of its electricity network such as information about power, voltage, current, etc. For that purpose is developed a server application called PSA.

PSA performs calculations and analysis of static load flow and tests over random changes in power systems, taking into account needs of liberalized electricity market. In addition to the basic tools to load the input data elements and edit the network model, within PSA application there are tools for automatic testing of random changes, calculations for the network transmission capacity (*NTC*), for the calculations of power transfer distribution factors (*PTDF*), factors for maximum flows, for scaling of production and consumption, as well as to display the results of different calculations.

Each operator sends its transmission network model that consists of:

- all nodes with data of active and reactive power and voltage constraints,
- boundary (X) nodes, fictional nodes that are located in the areas of interconnection lines,
- branches of the transmission network or lines, with parameters and maximum allowable current,
- transformers with a rated primary and secondary voltage.

[1]Milos Gajic is with the TelventDMS D.O.O., Narodnog Fronta 25, 21000 Novi Sad, Serbia, E-mail: milos.gajic@telventdms.com

[2]Marko Djukovic is with the TelventDMS D.O.O., Narodnog Fronta 25, 21000 Novi Sad, Serbia, E-mail: marko.djukovic@telventdms.com

[3]Sasa Devic is with Telvent DMS D.O.O., Narodnog Fronta 25, 21000 Novi Sad, E-mail: sasa.devic@telventdms.com

[4]Branislav Atlagic is with TelventDMS D.O.O., Narodnog Fronta 25, 21000 Novi Sad, E-mail: branislav.atlagic@telventdms.com

[5]Zvonko Gorecan is with TelventDMS D.O.O., Narodnog Fronta 25, 21000 Novi Sad, E-mail: zvonko.gorecan@telventdms.com

[6]Dragan Tomic with TelventDMS D.O.O., Narodnog Fronta 25, 21000 Novi Sad, E-mail: dragan.tomic@telventdms.com

Job of PSA application is to merge individual national network model in a comprehensive joined model, and to calculate the specific technical parameters that are later used as input to the auction software part.

PSA application must be able to perform calculations twenty-four hours a day, seven days a week. It is also developed client oriented desktop application intended for operators of certain countries. *PSA Client* allows a user to send national models, display input data, run specific calculation, get an insight into the results, search by preselected criteria and so on. Architecture of PSA software is presented in the Fig.1.

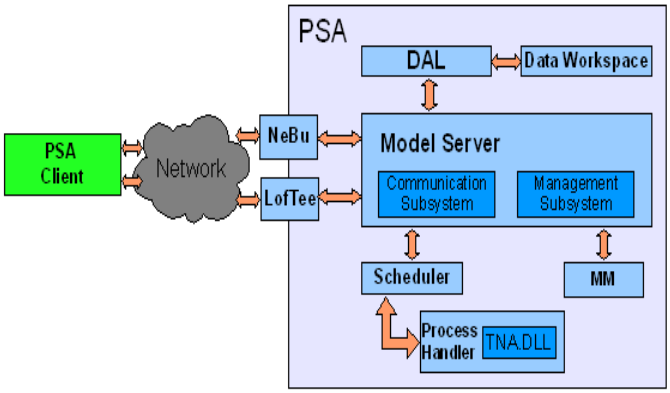

Fig. 1. Architecture of existing PSA solution

III. USED TECHNOLOGIES AND TOOLS

The latest technologies and widely used standards have been used for development of interoperational web services, web and mobile clients which can provide support for the above operations. The following is a brief description of the used technologies and tools.

- SOAP [1] is a protocol intended for exchange of structured data through web services over the network. It is based on XML as the format of messages. SOAP can provide a base layer for web services protocol stack, providing a foundation for building messages based on which web services can continue to build.
- WSDL [1] is language based on XML standard, which provides a model for describing web services. It defines services as collections of endpoints or ports.
- TCP/IP [2] (*Transmission Control Protocol/Internet Protocol*) is the most commonly used protocol for data transmission in computer networks. TCP is responsible for working with data in the transport layer, is suitable for the high-security communication channels of information transmission, but shows poor performance on the communication channels with frequent damage to the data (e.g. wireless).
- HTTP [2] (*Hypertext Transfer Protocol*) protocol is the most commonly used protocol for transmission of information on the web.
- SSL [3] (*Secure Socket Layer*), and it's successor TLS is the standard security technology for establishing a secure connection through unsecured network.
- WCF [4] (*Windows Communication Foundation*), the application interface within .NET environment for the construction of inter-related, service-oriented application. WCF is designed in accordance with the principles of service-oriented architecture (*SOA*), to be able to support the development of distributed client-server architecture. WCF web services have published WSDL interface through which any WCF client can request a service independently of the platform of services.
- Silverlight [5] is an application framework for writing and running Rich Internet Applications (*RIA*). The run-time environment for Silverlight is available as a plug-in for web browsers running under Microsoft Windows and Mac OS X. Silverlight is focused on streaming media, support for multimedia, graphics and animation, and give developers support for CLI languages and development tools. Silverlight is also one of the two application development platforms for Windows Phone. A free software implementation named Moonlight, developed by Novell in cooperation with Microsoft, is available to bring Silverlight (versions 1 and 2) functionality to Linux, FreeBSD and other open source platforms. General concepts mentioned here sound very familiar, because similar approaches were applied in previous technologies, like Java, ActiveX, ShockWare and Adobe Flash. The most successful technology of all is Adobe Flash, to which Silverlight is most a like. But, to make a head start over well proven competition, Silverlight aims to join the power of flash with development abilities that .NET framework has to offer. The advantage of Silverlight over Flash is usage of strong .NET framework and ability of writing code in pure C# programming language. Flash offers better designer support and larger user community, but without comprehensive development framework. To point it out, for any enterprise project requiring heavy programming or data access that would benefit from Windows desktop integration, Silverlight is the way to go.
- Microsoft Visual Studio is an Integrated Development Environment developed from Microsoft. It can be used to develop console and graphical applications, web sites, web applications and web servers in a controlled an uncontrolled code (manage and native) for all platforms supported by Microsoft such as: Windows, Windows Mobile, Windows CE.
- Eclipse IDE is an Integrated Development Environment created by Open Source community. It that can be used for developing Java and Android application. It provides superior Java editing with validation, incremental compilation, cross-referencing, code assistants.

IV. IMPLEMENTATION

In this section we will try to present developed clients for PSA application, how system architecture is expanded in order to retain high system stability and usability, with remarks on base client – *PSAClient*.

A. Silverlight software solutions

The beauty of Silverlight is its easy joining with WCF, the technology of application server side. Developers are enabled to, in a few simple steps, setup a communication channel with WCF web services and almost all capabilities that this technology has to offer. Silverlight communicates with server side in asynchronous way, which means no application freeze on waiting for reply. In our Silverlight web client we will use a communication protocol that allows us bidirectional communication between client and server. For those reasons Silverlight was a simple choice as technology for browser client.

Since the current solution is still under development, and since neither *LofTee* nor *NeBu* web services do not allow changes that could potentially compromise the functionality of PSA Client, it is created a new web service middleware named *LightTube* (Fig 2.).

LightTube web service was introduced to bridge the gap between existing solutions and its expansion in the form of a new, Silverlight client. It serves as an intermediary between *LofTee* web services and Silverlight clients. *LightTube* communicates with the web service through a protocol not supported by Silverlight, and provides services to clients through protocol that is supported. That is why its introduction was necessary. Communication with web service is accomplished by simple adding a reference to the service, which generates a proxy object (the client), through which the transport takes place. All information to generate client are published through the WSDL view. Silverlight client is visible to users of the system. It is used to display the model, run certain processes, information on the progress of running progress, and for viewing results. The application supports the initiation of load flow calculations (*AC Load Flow* and *DC Load Flow*). Load flow calculation is the basic calculation in the auction process. Seventy percent of all other calculations refer to the results obtained from load flow calculations [6].

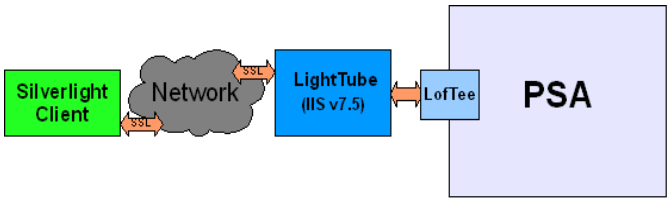

Fig. 2. Silverlight architecture

We have chosen Silverlight for developing client side application because of the following advantages:

- very good design capabilities,
- easy learning,
- good compatibility with existing system,
- no-cross browser issues,
- cross-platform options.

B. Android and Windows Mobile Software solutions

Choosing right technology for application in fast growing and fast changing mobile market can be a tough job to do. In our development process we chose to use two presently leading platforms in this area: Windows Mobile as natural choice considering technology of PSA server side, and Android which is an absolute market leader.

In order not to endanger the functionality of the auction server, and to successfully connect customers with mobile phones, it is developed PSA Adapter, an application presented in Fig 3.

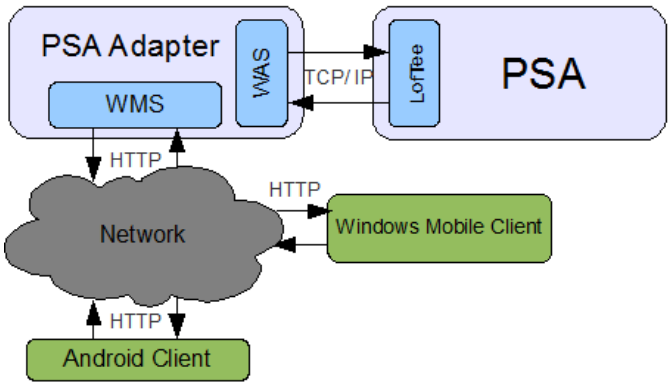

Fig. 3. PSA Adapter

Adapter establishes a connection with the auction server and presents certain information to clients who access it using mobile phones based on Android and WM operating systems. Android client application is written in Java programming language, while WM client application is written in Visual Studio and based on .NET framework environment – same as PSA Adapter. Communication between the auction server and PSA adapter is established via TCP/IP protocol. PSA Adapter is an application written in C# which consists of two web services. First, *WMS* web service, is used for communication between mobile phone and PSA Adapter, while second, WAS web service, is used for communication with the auction server via *LofTee* web service. The need to introduce an intermediary in communication between the auction server and mobile device is implemented for following reasons:

- auction server is implemented to support only connection with clients via TCP/IP, while mobile phones require the HTTP protocol,
- data structures that are used for transmission between the auction server and clients are complex and large mobile devices,
- there was no possibility of modification on the server side because of its functionality.

PSA Adapter takes complex data structures from the auction server, perform their filtering and forwards simplified and desired structures to clients that will be displayed on the mobile phones. This approach enables mobile applications to work faster because all data is prepared in the PSA Adapter so that the network transmits only minimal amount of information. The transfer of information between the PSA Adapter and the auction server is implemented in both directions using WCF duplex services. That is a kind of communication protocol between client and server side, where client can invoke communication with the server (by calling corresponding method), and after that, in similar way, server can invoke methods on the client side. Communication

between *WAS* and *Loftee* is established as user sends request to PSA Adapter, which is then forwarded to auction server. Server processes the request and creates an output structure that is returned to the adapter which receives data and performs repackaging into a format suitable for sending to the mobile phone.

V. Performance Results

The data needed to perform the auction process is given in the form of the following files:

- UCT (*Union of the Coordination of Transmission of Electricity*) file contains information about the status of a regional electricity network (nodes, transformers, branches),
- CBCO (*Critical Branches Critical Outages*) file contains information about the critical sectors of the system as well as critical network outages,
- ANTR (*Already Nominated Transactions*) file contains information about the amount of power that is exchanged between the operator through dealer,
- AATR (*Already Allocated Transactions*) file contains the electric transmission rights that are allocated in a previous stage, and the transfer of rights in case of cross – border exchange.

Table 1 shows the performance of applications described in this paper. Values obtained for the mobile phones have been calculated using simulators. The test was performed on a model that contains 7707 nodes, 6000 branches and 1500 transformers.

TABLE I
PERFORMANCE OF GIVEN APPLICATIONS

Operation		PSA Client	Silverlight	Mobile	
				Android	Windows
Making connection		0.8 [s]	1.5 [s]	2.7 [s]	3.1 [s]
Loading	UCT data	5.5 [s]	49.8 [s]	241.7 [s]	249.8 [s]
	CBCO data	0.6 [s]	2.1 [s]	2.7 [s]	6.7 [s]
	ANTR data	0.5 [s]	1.6 [s]	1.9 [s]	3.3 [s]
	AATR data	0.2 [s]	1.2 [s]	2.1 [s]	3.1 [s]
Starting calculation		0.2 [s]	0.97 [s]	1.3 [s]	1.7 [s]
Getting results of LF calculation		1.7 [s]	5.7 [s]	29.2 [s]	32.9 [s]
Searching	Node	0.3 [s]	1.2 [s]	1.6 [s]	2.2 [s]
	Branch	0.2 [s]	0.9 [s]	1.2 [s]	1.6 [s]
	Transformer	0.1 [s]	0.5 [s]	0.7 [s]	1.1 [s]

VI. Conclusion

We can conclude that each technology has its advances and drawbacks. Android and WM are oriented to fast growing mobile market, but their performance are noticeably lower than they are for Silverlight, which is designed primarily for personal computers.

Here, in mobile applications, we may notice slightly better results for android solution, but that should be taken with caution. We must keep in mind that developing Android application aimed to communicate with Windows services is not even nearly as easy as developing WM application aimed to communicate with Windows services. It is questionable that valuable time lost in development process is worthy slightly better performances. On the long term that is true, but in today world, if solutions are demanded for the next day, it's other way around.

Silverlight can also be used in developing mobile applications, but in environment with much less resources, it is questionable in what degree would that affect its performances. Main power of Silverlight is its capability to interact with machine it runs on, to use hardware to boost its performances. Silverlight is maid to be a strong client side application, which requires a sizable amount for resources just to run itself. Therefore, we should use it for developing web application that tends to be as powerful as desktop applications. To sum it all, main guides when choosing the right technology for any future application we plan to develop should be:

- In what environment will it be running?
- How much time we have to develop it?
- How familiar are we with desired technology?
- How that technology fits in our current system?
- What performances are expected? And of course:
- How big is user community for desired technology?

Today, many technologies in area of web development are rising and falling, and choosing the right one is just getting harder then it has ever been.

References

[1] Prof. dr. Branko Milosavljević, *XML i web servisi.* Fakultet tehničkih nauka, *Novi Sad, 2007/2008,* http://informatika.ftn.uns.ac.rs/XWS

[2] Andrew Troelsen, „Pro C# 2008 and .NET 3.5 Platform, Fourth Edition", Apress, 2008

[3] Prof. dr. Branko Milosavljević, *Bezbednost u sistemima elektronskog poslovanja.* Fakultet tehničkih nauka, *Novi Sad, 2007/2008,* http://informatika.ftn.uns.ac.rs/BSEP

[4] Chris Peiris, Shawn Cicoria, Amit Bahree, *Pro WCF: Practical Microsoft SOA Implementation.* Apress, januar 2007

[5] Matthew MacDonald, *Pro Silverlight 3 in C#.* Apress, novembar 2009., http://apress.com/book/view/1430223812

[6] Prof. dr. Nikola Rajković, Analiza elektroenergetskih sistema II. Akademska Misao, Beograd, 2008.