Web Service Standards and Real Business Scenario Challenges.

Conference Paper *in* Conference Proceedings of the EUROMICRO · January 2003

DOI: 10.1109/EURMIC.2003.1231627 · Source: DBLP

3 authors, including:

Ayse Bener
Ryerson University
186 PUBLICATIONS **3,223** CITATIONS

SEE PROFILE

Haluk O. Bingol
Bogazici University
40 PUBLICATIONS **699** CITATIONS

SEE PROFILE

Web Service Standards and Real Business Scenario Challenges

Çiğdem Patlak, Ayşe Başar Bener, Haluk Bingöl
Department of Computer Engineering, Boğaziçi University
cigdempatlak@yahoo.com, {bener,bingol}@boun.edu.tr

Abstract

The Web services paradigm is expected to transform the Web into a distributed application-to-application network. The Web services landscape is in an evolving state with core specifications almost mature and gaining widespread acceptance. Yet for some specific Web service concepts in-development specifications exist. Despite this fact, organizations should start to utilize primary Web service standards to become familiar with their distinguished opportunities in connectivity and interoperability. The principal purpose of our study is to illustrate the challenges of Web services adoption for real business use cases by a sample financial Web service scenario and solution.

Keywords
Web services, financial Web services, ATM

1. Introduction

Web service concepts already promise to hold a massive potential for distributed computing and application integration, however to reach up to the complete vision of Web services is still a long-term issue. Today many Web service examples have API-centric features and demonstrate client-server like communication operations, whereas B2C Web services across enterprises are not commonplace. Currently many companies prefer to deploy internal Web services and hesitate in adopting them for cross-organizational applications outside corporate firewalls before standardized security roadmaps have been set. There are several issues, which are agreed upon, to be in need of immediate improvements for enterprise-robust Web services[1]: Security/Privacy, Messaging/Routing, Quality-of-Service (QoS)/Reliability, Business Processes and Transactions, Management, Performance, Interoperability. Furthermore vendor-specific incompatible solutions might arise due to the increasing number of non-standardized specifications in some fields of Web services. These shortcomings are chief barriers to widespread external usage. In our belief, the real impact of web services can only be measured when large deployment of this emerging technology takes place. Until then, for a so-called warm-up, making use of principal standards such as SOAP, WSDL, XML is the most advisable approach. For that reason, we have worked on typical financial use cases to produce experimental Web service scenarios.

This paper is organized as follows. Section 1 consists of brief information about Web services, the motivation of our research and distributed computing in the Finance sector. Section 2 deals with a sample Web service for a financial use case and our observations. In section 3, we place our use case into an outsourcing solution and list the challenges that occur when Web services are integrated into much more sophisticated environment settings. At the end, conclusions and final remarks of our studies can be found.

1.1. Web Services in the Finance Industry

Financial institutions offer a variety of services and follow leading trends in IT technology. As financial service processes vary in duration and significance, it is possible to create a diverse range of usage scenarios based on such services and identify the current state of Web service adoption. Some financial services include instant services that use real time data, whereas others are made up of long-running transactions, enterprise business processes within the organizational workflow. It is also expected that financial services will lead the way for externally deployed Web Services [3], so that more sophisticated business needs can also be analyzed.

Many banks keep in track with the latest technology to provide a diversity of new services and products to various customer groups, such as Internet banking, credit card products, ATM services, phone banking, call center services, etc. Multi-channel services, new products offerings or ever-changing business operations require flexible and interoperable system architectures

Proceedings of the 29th EUROMICRO Conference "New Waves in System Architecture" (EUROMICRO'03)
1089-6503/03 $17.00 © 2003 IEEE

or steady support for routine reorganization of IT infrastructure, where the latter choice is not preferable in the competitive financial market. Most companies already use the Internet to deliver their services and products online and utilize Internet standards, such as TCP/IP, HTTP, HTML, XML, J2EE, .Net – the building blocks and principal players of Web services. Therefore a shift towards Web service methodologies is not a decision that gives rise to massive changes in the technological architecture of such a company. Web services provide several bridging and integration techniques for designing and implementing loosely coupled and compatible applications on distributed software domains. In case of multi-channel financial services, platform-independent web services can reduce deployment and development time and cost by a reasonable extent Moreover, client back-end operation and presentation logic can be customized for each channel separately and independent of the Web service's implementation.

2. Financial Service – Use Case

The motive throughout the study was to identify the challenges of Web services adoption for real business use cases in a distributed computing environment. Therefore first a typical banking application with distributed computing features has been selected to produce an experimental Web service scenario and solution. The overall scenario of the sample financial service includes an enterprise that wants to implement a business application for processing credit card purchase orders online using a set of Web Services. The purpose is to simplify implementation and provide a "plug-in" service that enables service requestors (merchants) to increase the functionality of their systems with minimal systematic changes.

The actual application architecture transfers data in special format XML-files. The principal drawback of the original application is the extensive use of XML-parsing performed, which can become a significant performance penalty. Moreover, the application interface is not applicable for complete reusability across internal and external services without any customizations.

The plan is to wrap the online credit card validation and real-time payment authorization service with Web service features. A wide range of products automating Web service standards coding exist. Based on the sample scenario, an authorization request, sent directly by a merchant to the bank, contains specific parameters such as terminal-id, transaction- amount, card-number etc. A brief overview of scenario steps is made in the lines:

- A merchant with a virtual POS installation, which communicates via HTTPS, sends an authorization request message to the Payment-Authorization Web service using SOAP.
- After the Payment Authorization Web service has been invoked, field-level control checking is performed.
- According to the card-type (international, domestic, on-us), authorization requests are processed.
- The response for an authorization request is returned through the Payment-Authorization Web service.
- The merchant displays, prints or logs the authorization response information as needed.

The new Payment-Authorization Web service is self-describing. It can easily be integrated with internal applications that handle authorization requests from branches or mobile payment applications, without the development of several client interfaces for varying service calls. The client application's presentation logic is not dependent on the authorization Web service. Due to security considerations, merchant and bank communication is established through HTTPS. A further security aspect is the control of clients' IP addresses during Web service invocation. The current experimental system architecture does only permit access to On-Us merchants for the Payment-Authorization Web service by using static IP addresses.

3. Extended Financial Service

In general, the payment processing service could be considered as a basic financial operation based on a request response mechanism, thus resembling initial Web service examples. Yet the sample application exists within a dynamic computing infrastructure and is a real-time process, which has to be completed within seconds (max. 1-2 seconds). In addition, it is a mission-critical application as it directly triggers accounting programs and reports authorization results and response times. Conclusively, the authorization application requires full system integration for any kind of programmatic changes. But to picture the interaction of multiple services in a distributed system, an advanced financial scenario, which deals with "ATM Networks", has been established.

There is a trend towards outsourced ATM-networks [7]. Certain firms provide ATM services to multiple banks based on special fee-payments. If all ATMs within such a network are Web-enabled, there is no

obstacle left to make use of Web services. The following section summarizes the scenario:

- There are 99 banks in Finance-Country, named Bank1, Bank2 … Bank99 respectively. Finance-Country has over 50 cities. Each bank has its own SW/HW infrastructure, exploits different banking, ATM, IVR etc. applications, but specific service offerings are all wrapped up with Web service features.
- Company Money-Maker sets up an ATM network in Finance-Country. It serves each bank in return of a certain fee amount. Money-Maker ATM's provide usual ATM operations by accessing financial Web services provided by member banks.
- If a customer uses a Money-Maker ATM, the ATM uses the card bank identification number (BIN) to identify a specific member bank and to display the original bank HTML page, which simplifies financial activities for customers. Input parameters that are used for redirection and service access are personal identification number (PIN), operation type (Query, withdrawal, deposit, etc.) and transactional parameters (e.g: amount). Local HTML pages might include a URL as a Web service link that redirects operations to the bank Web server. Such service calls require data encryption and authentication.
- The Money-Maker ATM offers a unique service-list, but in fact there are 99 banks that offer their own original service. These 3rd party ATMs performs necessary page updates from bank Web servers during their idle loop.

Besides Web services, other application integration techniques can be taken into consideration for this project. But the ATM network idea is a profound use case to go through Web service transformation and its challenges on a step-by-step basis.

3.1 Advantages of Web Service Usage in the "ATM Network "

In general terms, standardized application interfaces and reusability are the most distinctive features offered through Web service usage in our ATM network scenario. The 3rd party ATMs do not differ according to varying bank host architectures. Formats for transactions, functions, etc. can be retrieved dynamically from each host. Operational changes such as transaction order or sequential organization are performed on the host side. Only local HTML changes take place client side. Certainly, such a solution promotes a host and dummy-client architecture, which can simplify integration of alternative distribution channels, such as kiosks, mobile banking, Internet banking, etc. If joint page design is applicable, despite differing monitor sizes of a PC, mobile phone or ATM, customers might use the same visual interfaces for financial operations. The WSDL document forms the basis for such a dynamic and self-descriptive integration structure; changes are reflected in the description file so that only accessing the web service definition is required to start rapid adaptation. If the ATM network provider decides to serve another bank, the ATM only gets information about a new WSDL document and its location.

Web-enabled ATMs make extensive use of Internet standards such as HTML. The key benefits are simplified and user-friendly client interfaces, development simplicity and cost reduction, interactive screen support, multimedia and plug-in support. Interactive multimedia interfaces offer a wide range of advertisement opportunities that attract more customers; an ATM idle loop is suitable for advertisement display. Furthermore, HTML pages can be stored on a Web server and dynamically retrieved. Such remote page access is simple, fast and cost efficient. As a result, local page storage on ATMs is done at startup and is repeated only if updates took place on the server side. There are already some solution offerings on the market [7], which provide functions like card-reader, cash-deposit functions etc. as ActiveX objects.

3.2. Specific Considerations

Fundamental drawbacks that were identified during the ATM network study are related with security, performance, business processes and cost.

- Producing initial ATM passwords for host communication is a general problem. To supply RSA based passwords would be a solution, however Europay and VISA support this feature only in EMV compatible environments [8].
- For cost reduction, dynamic IP connections have been taken into consideration. But such connections pass company firewalls, what makes the system prone to DOS attacks. In any case, an Intrusion Detection System (IDS) is needed. Banks already make use of Web servers and produce online services, so that the presence of an IDS is certain.
- If ATM connections in the ATM network only use encryption without tunneling methods, the line might be sniffed. If enough information is collected, decryption of passwords is possible. Another possibility is that request packages might be caught. Without performing complete message decryption,

using only specific fields like a transaction number, network packages can be sent to the host over and over again. Continuous cash deposit is a basic example for such actions.

- In many cases ATMs and hosts do not communicate directly. There is an additional server handling ATM messages. For example, there can be an IBM host with CICS clients that need a CICS server for interaction. Since there is a plan to use Web services, a standalone or embedded Web service server is also needed between the host and ATM, which is used for Web service deployment, etc. The communication server might not be able to process XML based information and in that case the Web service server will parse XML messages and transfer them to this server placed in front of the host. A Web service server cannot be directly connected to the host. This is also a precaution so as to avoid client side changes whenever host-based changes take place.
- 3rd party ATMs are located outside the host firewall. Therefore these ATMs do not have direct access to bank servers and bank networks. On the contrary, banks locate their own ATMs within company firewall boundaries and the host itself is also behind the firewall. 3rd party ATMs have to use port 80, which is the only port open outside of the bank firewall. In conclusion, a new configuration or rearrangement of connections and security policies has to be established. Many banks are unwilling to make host side changes and want that the company providing the ATM network solution to deliver complete integration services. But in case of complete outsourcing, Web service adoption will make cost reduction difficult.
- During the analysis phase, services that are basically used for information retrieval were described as appropriate to be offered as Web services. Online stores or financial information providers were listed as potential Web service examples. But mainstream financial services were not put into this category, as financial Web service usage does not seem feasible and cost-effective to financial institutions.

4. Final Remarks

A starting point for web service adoption could be to make overall analysis of service-oriented and process-centric architecture concepts, before diving into the ocean of Web services standards without any insight. While industry-wide standardization efforts continue, companies, who plan to utilize Web service standards, should spend this time on creating custom-roadmaps for their Web services software development life-cycle. Aside from specialized feature requirements or standardization activities, there are only a few real business scenarios and implementations for specifying best-of-breed adoption methods and potential Web service benefits. Even our basic financial application that appeared to need a straightforward Web service enablement operation, demonstrated unexpected development considerations. Certainly a detailed integration plan would have accelerated the development phase. Except for early Web service adopters, many companies start with low-value internal Web service implementations. The extended financial use case provides a realistic picture of where we stand today with Web service business practices. Thus we believe that moving to mission-critical applications can only be safely taken into consideration, when higher-level Web service standards have matured and widespread usage of emerging Web services technology takes place.

5. References

[1] Bloor Research, “Web Services Gotchas, How Enterprises Can Build Secure, Reliable, Performance-optimized Web Services Solutions While Waiting for the Standards to Mature”, Bloor Research, North America, July 2002

[2] W3C –Web Services Activity, “Web Services Architecture Usage Scenarios”, W3C Working Draft, 30 July 2002

[3] Narsu, U. and P. Murphy, “Web Services Adoption Outlook Improves”, Giga Information Group, Inc., 2002

[4] Patlak, C., *Web Service Orchestration Standards*, M.S. Thesis, Department of Computer Engineering, Boğaziçi University, Istanbul – Turkey, 2003

[5] Maier, G.E., “The Impact of Web Services on Retail Banking”, IONA Technologies, 2002

[6] IBM, “Security in a Web Services World: A Proposed Architecture and Roadmap”, Security whitepaper, IBM Corporation and Microsoft Corporation

[7] ATM Marketplace, http://www.atmmarketplace.com/, 2003

[8] EMVCo, http://www..emvco.com., 2003

[9] Adams, H, D. Gisolfi, J.Snell and R.Varadan, "Best practices for Web services - series", IBM developerWorks Web services articles, 2003