Web Services And How the Technology May be Used to Secure the Homeland

I. Executive Summary

Sharing critical information, while protecting that information from prying eyes that could compromise national security and trade secrets, is a delicate balancing act. In order to do this, one doesn't necessarily have to create entirely new systems; rather, one has to understand legacy systems that are available and see if they can be enhanced to meet requirements. The popularity of using Web Services to do this by integrating data securely among disparate computing platforms is rising, due in part to economics. Web Services provides the ability to integrate disparate (legacy) systems, and cull meaningful data from them, without having to re-write them. Without a Web Services architecture, legacy system owners would have to work out a unique protocol to share data. This is expensive and time-consuming.

Web Services applications are nearly limitless. With Web Services employed, a police officer would know a person apprehended in a robbery had his visa revoked just this morning. Or, the origination of a deadly disease may be traced to canal water in upstate New York. Or, a shipping container holding a biological agent may be detected prior to entering a U.S. port.

The purpose of this paper is to provide a mini-tutorial on Web Services, and suggest ways in which this technology may be used to secure the homeland. DHS sifts through tons of unstructured data; this takes time, and the sheer quantity of information impedes efforts to see links or draw big-picture conclusions. The Web Services mini-tutorial defines and describes how this technology works, its goal, standards, the promise, benefits, and security requirements.

There are various ways in which Web Services may be used to integrate data for use as meaningful information to help secure the homeland. We drown in data; as a life jacket, we should connect data from different systems in order to easily find and examine pertinent information. Linking disparate systems is not a trivial exercise, and requires considerable time and effort to design and maintain processes to support multiple data formats and protocols, for both intra and inter-enterprise communications. Web Services provides a framework for real-time integration using open standards over the Internet, which has a lower deployment overhead than traditional integration technologies. Continue reading to learn more about Web Services and how it may be used to protect our homeland.

II. Web Services Tutorial

Definition

Not surprisingly, given how rapidly the sector is evolving, there is no universally accepted definition of "Web Services." Research reveals varying definitions. The Gartner Group defines Web Services as "Loosely coupled software components that interact with one another dynamically via standard Internet technologies." In contrast, Forrester Research defines Web Services somewhat abstractly as "Automated connections between people, systems, and applications that expose elements of business functionality as a software service and create new business value."

Put into more tangible terms, Web Services are building blocks for creating open distributed systems. At the most basic level, Web Services can be considered a universal client/server architecture that allows disparate systems to communicate with each other without using proprietary client libraries. Web Services are a collection of functions that are packaged as a single unit and published to the network for use by other software programs. Web Services are loosely coupled, encapsulated components that accomplish a well-defined purpose and make their interfaces available via standard protocols and data formats so they can be invoked over an internal network or externally, via the Internet (webMethods, Demystifying, p. 5). Simple examples of Web Services include a credit checking service or a package tracking service. Web Services can also aggregate other Web Services to provide higher-level functionality—for instance, a complete order management service, in true business process collaboration (webMethods, Implementing, p. 6). Functions range from the simple (e.g., mortgage calculator) to the complex (a collection of services that processes payments) (Gallagher).

So, why do we need Web Services? What will it do for us, and how? The Web is not the monoculture it once was. Powerful PCs are mostly underutilized, having been reduced to the role of dumb terminals running Web browsers. The ultimate purpose of most Web sites is to present the end user with a screen, or dead pages. But what if, when you log on to a Web site, your software sends out an executable message that allows for true interaction with the site? For example, if you are looking for information on solar greenhouses, instead of a Yellow Pages listing of companies making them, you receive interactive training tools hat help you set up a page with a pricing schedule for the best options from leading manufacturers (Miles).
Or, what if, using Web Services, you set up an enterprise Web site that provides in-house users with behind-the-firewall product information, customers with spec sheets and order forms, and business partners with site-by-site parts information so that your inventory would be automatically refilled by the right parts provider? This new type of super Web is what Web Services proponents promise (Miles).

Goal

The goal of Web Services is to provide a common interface that allows computers to run programs, share data, and access services with loosely coupled, Web-based middleware (Conry-Murray). Ultimately, the vision of Web Services is to facilitate and automate process collaboration, both inside and outside the enterprise, and to eliminate inefficiencies typically caused by the unavailability of timely information and by non-value added processing steps such as data re-keying (webMethods, Implementing, p. 11).

Enabling technologies and standards

Web Services needs standards in order to work as advertised. Most industry Web Services participants agree that Web Services solutions developed by the World Wide Web Consortium and other standards bodies should incorporate at least the following: Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI) (Miles). These are defined in more detail below. From a technology point of view, Web Services is about connecting disparate software systems using a set of standard protocols in a way that enables seamless information and process flow among the different systems (webMethods, Implementing, p. 7).

XML

XML is similar to Hyper Text Markup Language (HTML) that is used on most Web sites, and it will be the lingua franca for Web Services. It is ideal for sharing information over the Internet among different applications and businesses (Miles). XML consists of a set of user-defined tags that provide a common definition of data elements in order to facilitate data exchange across heterogeneous computer systems (Miller). XML is a standard language used to structure and describe data that can be understood by different applications. XML enables diverse computer systems to share data, regardless of operating system and programming language. Finally, XML utilizes HTTP as a transport, allowing remote method requests to pass through enterprise firewalls via standard ports, such as port 80 (DISA, p. 49). XML allows organizations to do business with each other by bridging the differences among each organization's individual proprietary systems (Tanner).

SOAP

SOAP is a lightweight, XML-based protocol for exchanging information in a centralized, distributed environment. Simply stated, SOAP is an envelope structure that carries an XML message, describes what is in it, and how to process it. It includes a set of encoding rules for instances of application-defined data types and a convention for representing remote procedure calls and responses (Miles). It is completely independent of the underlying transport protocol, so SOAP message can be exchanged over many transport protocols (webMethods, Demystifying, p. 5). SOAP is used to send XML-based encrypted commands and messages. A SOAP message is the envelope in XML and sent via HTTP (unencrypted port 80) (Gallagher). It thus inherits the security holes common to HTTP implementations. SOAP transactions/messages can be strongly protected through digital signature and encryption (DISA STIG, p. 51).

WSDL

WSDL is similar to XML. It treats network services as a set of endpoints operating on messages about documents or procedures. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are then combined into abstract endpoints, or services (Miles). WSDL is a specification defining how to describe Web Services in common XML grammar. WSDL is a vocabulary for describing what a Web Service does and what commands and data it will accept. A WSDL document provides all the information needed to access and use a Web Service, describing the Web Service's interface, how it communicates, and where it resides (webMethods, Demystifying, p. 5).

UDDI

UDDI, a global registry of Web Services, is an essential search tool. Version 3.0 is the latest registry specification, supported by more than 200 members of the UDDI consortium (www.uddi.org). UDDI gives organizations and Web Services a universal way to convey information over the Internet (Miles). UDDI is essentially a "yellow pages" into which Web Services and their associated descriptions (in WSDL) can be registered and retrieved (webMethods, Demystifying, p. 5). UDDI is a principal XML-based protocol that is the discovery layer within the Web Services protocol stack (DISA STIG, p. 51). UDDI is a specification for creating directories of Web Services. In addition, UDDI describes a protocol for publishing and discovering information about business services, and is not a required element for implementing a Web Services architecture. However, as organizations realize that there are significant benefits in moving to a service-oriented architecture, they will realize that UDDI is no longer an option, but a necessity (LaMonica).

The Promise

Web Services promises to revolutionize business process collaboration by enabling the next generation of composite systems. The following characteristics drive Web Services' potential:

Interoperability

Web Services can be written in any language, and any Web Service can interact with any other Web Service. Interoperability among disparate software systems is not a new challenge, but many past attempts at solving the problem have met with limited success because of a lack of standardization or broad support.

Industry Support

Web Services address the interoperability issue by having the widest industry support from the start. Every major software vendor has endorsed Web Services, the SOAP standard and surrounding Web Services technologies.

Ubiquity

Just as industry support is key to interoperability, interoperability will drive the ubiquity of Web Services. Furthermore, since Web Services communicate using basic protocols such as HTTP

and XML, any platform that supports these (highly common) technologies can both host and access Web Services.

Low-Entry Cost

The broad support for Web Services, coupled with the relative simplicity of the underlying technologies and the availability of development toolkits, has made cost-effective implementation possible (webMethods, Implementing, p.6-7).

Web Services interface standards decouple the caller of a service (i.e., DHS) from the service's implementation platform, run-time environment, and even its physical location (i.e., International Shipping Port Organization). This independence creates the flexibility to change not only the implementation and the location of a Web Service, but also the provider itself, without adversely affecting the higher-level business application.

Benefits

There are a number of inherent advantages associated with employing Web Services. The common attribute of all of them is that they save money. The most important one is the ability Web Services afford to communicate with disparate systems securely and efficiently. By adhering to the Internet standards such as HTTP, XML, and SOAP, development of Web Services applications can be accomplished on disparate platforms using a wide variety of development languages and tools, without having to change or reconfigure existing organizations' applications. The great advantage of using Web Services technology is the cost reduction it provides. Automating process saves money, and Web Services is a cheap way to do that (Gallagher).

Integration is consistently cited as being the chief concern for organizations being able to use their existing information systems to share and correlate information. XML eases this complex integration process.

A Web Services solution can post enhancements to the main system without disruption in service and without having to port software to organizations' locations. Changes to the application are seamless. Implementing Web Services also provides from a single platform solution, reduced cost and risk, faster time-to-benefit, and forward capability.

Security Requirements

Not surprisingly, Web Services have similar security needs to almost any other Internet-based application. Security is even more critical with Web Services since they may expose an organization's internal operations. In general, security for Web Services seeks to provide authentication, authorization, confidentiality, and integrity. Existing secure web standards, such as HTTPS and SSL, are not able to address XML specific issues such as partial document signing and the fact that XML documents are often processed in stages along loosely-coupled

network paths. To solve these problems, developers use XML Encryption to encode individual parts of the XML document; XML Signature to manage the integrity of XML as it moves through the Web, again along loosely-coupled network paths; and XML Key Management Specification to deal with Public Key Infrastructure (PKI) verification and validation (DISA STIG, p. 49).

Web Services technology used to integrate data/systems is very promising; however, it takes people to make sure that the systems are working securely. Security standards groups are addressing many of the security features with the promise of interoperability, while others require careful implementation considerations and solutions and may not be interoperable (webMethods, Requirements, p. 19). It would be very inappropriate to not consider security for any Web Services project, let alone a DHS project; security must be a critical part of any Web Services implementation! Another note of caution is that because Web Services' customer-driven requirements and site operating environments are varied, a cookie-cutter approach to Web Services security is not practical (DISA STIG, p. 5).

III. How Can Web Services Help to Protect the Homeland?

Homeland Challenges

One of the toughest challenges with the many homeland security-related information sharing and collaboration projects is getting senior managers to throw their weight behind the interagency initiatives (Zyskowski). There are also technical issues to address, not to mention the continuous threat that looms over us. To quote Robert Dacey, General Accounting Office Director of Information Security, "We don't know exactly what threats will be tomorrow, but there are things that can be done to minimize or mitigate the impact (Walker)."

Web Services' Role in Protecting the Homeland

Following are various ways in which Web Services may be used to protect the homeland.

International Shipping Container Tracking System (ISCTS)

Many international shipping containers entering the United States are not being inspected. There are many disparate systems that track this type of information. We would suggest that the DHS create one Web Services system that tracks international shipping information (origins, contents, employees, etc.), and correlates it with customs information to safeguard our ports. For example, a flag would be raised if records indicate that a boat from Scandinavia is carrying bananas, when bananas do not even grow in Scandinavia. Customs agents could receive electronic manifests of cargo contents prior to ships arriving at U.S. ports. The goal is to identify shipments that could be a cause for concern and focus on those shipments.

Bordering Nations Information Exchange System (BNIES)

According to Bill Sheppit, Immigration Counselor with the Canadian Embassy in Washington, D.C., "Border security is probably the single most important policy issue facing Canada in the

next five years (Hasson and Michael)." The DHS could use Web Services technology to exchange information to manage numerous immigration services, such as visitor visas, temporary worker visas, and foreign students and refugees. Photographs and fingerprints of foreign visitors arriving at U.S. airports should be captured for data coloration.

The use of Web Services would achieve economies of scale in that the information would be gathered once and used many times. "I guess all I'm asking is when can we get effectively engaged so that the sooner we can start collaborating, the more our dollar can be shared to collect something once and then share the information," per Jim Hamilton, Director of Information Resource Management at Ontario's Ministry of Natural Resources (Sarkar and Holmes). Today there is no way to know when, or if, visitors leave the U.S.

National Alert System (NAS)

The 9/11 disaster drove home the importance of being able to issue emergency alerts to the public quickly and efficiently. The need for technology that connects local, state, and federal agencies and allows them to share critical information and images in a timely fashion has become very clear. Rosita Parks, Chief Information Officer at DHS' Federal Emergency Management Agency (FEMA), said that the agency is taking an inventory of what it has and what alert systems it could use to warn the public in the event of an emergency. Web Services may be used to provide a common (national) vision to law enforcement allowing them to respond quickly and efficiently when there is danger of loss of life. The NAS could collaborate a wide variety of threat scenarios and send alerts to multiple locations in multiple states to inform the public of national, state, and local threats and alerts.

Medical Threat Integration System (MTIS)

MTIS is a Web Services solution that could aggregate patient information from all clinics, hospitals, and offices. It would be an integration system that allows hospitals, 911 call centers, pharmacies and employers to collaborate on potential outbreaks of bioterrorism-related diseases. A Health Alert Module (HAM) could pull certain health data from health care provider systems to create a national system that recognizes suspicious disease outbreaks more quickly, and alert proper officials. The systems would be designed to indicate the presence of a bioterrorism agent, a naturally occurring disease outbreak, or a rare disease such as severe acute respiratory syndrome (SARS). Furthermore, this system could then be integrated with the Water/Air Quality Tracking System (WAQTS) described below to verify if the outbreak could have come from the water or air supply.

Water/Air Quality Tracking Systems (WAQTS)

The purpose of the WAQTS is to tap into various water supply and air quality systems to track contamination levels, water conditions, air pollutants, etc. Trends would yield important findings, and early warning notifications could be sent to appropriate authorities in the event of an adverse result to help in avoiding grave conditions, such as kidney attacks, death, and avoid to suffering. WAQTS would be used to make sure water is safe for public consumption.

"If a local water resource inspector in upstate New York finds an unknown toxin in canal water, how does that individual on the edge of government know how to lay his hands on the right

people … to know who to collaborate with in those circumstances at that time?" asked David Gilmour, CEO and founder of Tacit Knowledge Systems Inc. (Havenstein). Our integrated WAQTS would be able to send his request across a wide variety of agencies and match someone who is a good fit for that request. If every day or 10 times a day, transactional systems track maybe the pollutants in the water or quality of air, [the database] can run reports and track trends with metrics put in place to advise of changes to the quality of the water or air (Havenstein).

The DHS' Integration System (TDIS)

Charged with patrolling borders, analyzing U.S. intelligence, responding to emergencies and guarding against terrorism, among other tasks, the DHS mission is overwhelming. Today DHS is wading through vast amounts of "unstructured" data (news reports, memos, etc.) flowing into their systems. The government is inundated with raw data today: field reports, emails, collection summaries, immigration records, open source news feeds, etc. Some believe that by integrating the 22 DHS agencies and by combining information from other outside sources, DHS will be dealing with the largest volume of data in history (Essential). The TDIS, to be built on Web Services, will comb through the vast amount of unstructured data (from disparate systems) in its original language, organize it to make it useful, and present the information that is of the utmost importance to those who need it. Rather than build a customer setup for each agency and department, existing Web Services modules could be snapped together in Lego-like fashion (Gralla).

Employing Web Services to integrate government agencies is a good, and needed, initiative. Although two 9/11 hijackers living in San Diego were on watch lists, the CIA and FBI did not share information that could have prevented them from carrying out their attacks, which was just one of the many holes in the U.S. security system discovered after 9/11. "Databases used for law enforcement, immigration, intelligence, public health surveillance and emergency management have not been integrated in ways that allow us to 'connect the dots," Cooper told the House Government Reform Committee. "To better secure the homeland, we must link the vast amounts of knowledge residing within each government agency while ensuring adequate privacy (Hasson) and security.

IV. Conclusion

Web Services technology is clearly emerging as a way to offer enterprises an unprecedented opportunity to interface applications and services, both internally and across enterprise boundaries, using standard protocols. This paper presented a mini-tutorial of Web Services key concepts, and explained how the technology may be used to secure our homeland.

Web Services represents a significant step in the evolution of distributed systems and promises to transform computing by providing universal interoperability between applications running on different platforms.

Our weak economy has had a dramatic impact on the level of computer application and infrastructure investment. Organizations are not interested in investing in new applications. To

get more out of what exists, there is a need to integrate and align applications. Budget requests involving Web Services technology can be backed up by effective business cases. The value of Web Services comes not from building a brand new system, but by capitalizing on resources and skills already in place. Employing Web Services technology will help DHS build more effective IT systems at lower cost.

Former Governor, Tom Ridge, and his department, face many challenges in the mission of protecting the homeland: He has his hands full with transforming cultures, protecting information, establishing legal policies, and increasing intelligence training and education. We hope that the above information and suggestions help to lessen some of the issues raised by employing technology. Web Services provides a better "glue" for integration. Web Services will be the glue that brings together not only the 22 agencies, but also all of the other critical information sources, to exchange data and put it into a context that will provide valuable information flow.

References

Conry-Murray, Al (2003, January 6). Web Services Security Specifications. Network Magazine.

Defense Information Systems Agency (DISA) (2002) Web Services Security Technical Implementation Guide (STIG) . (Version 3, Release 1) [Guide].

"Essential Solutions for Homeland Security." (2003, Feb 1). Bitpipe, retrieved on 10 June 2003 from www.bitpipe.com/data/detail

Gallagher, S. (2001, October 29) Web Services. Baseline Magazine, retrieved on 25 June 2003 from www.baselinemag.com

Gralla, P. (2003, May 19). Web Services In Action. ComputerWorld.

Hasson, J. (2003, May 26). The first 100 days. Federal Computer Week.

Havenstein, H. (2003, June 2) Collaboration takes new forms. Federal Computer Week.

Holmes, A. and Sarkar, D. (2003, 16 June). Federal Computer Week, p. 8.

House Approves $29.4B for Homeland Security Dept. (2003, June 25). The Public Opinion, p. 10A.

LaMonica, M. (2003, May 20). Web Services 'yellow pages' gains ground. CNET News, retrieved 10 Jun 2003 from www.news.com.com

Michael, S. and Sarkar, D. (2003, 16 June). Federal Computer Week, p. 8.

Miles, J. (2003, June 2). Will Web Services Deliver? Government Computer News.

Miller, R. (2003, June 2). Starting a healthy dialogue. Federal Computer Week.

Security Overhaul Ordered at Nuclear Weapons Labs. (2003, June 25). The Public Opinion, p. 10A.

Tanner, J. (2002, Feb 1) Having it both ways; transactions on the edge. Telecom Asia, retrieved 15 June 2003 from www.telecomasia.net/telecomasia/article

Shipley, G. (2003, June 26) Are you vulnerable? Network Computing.

Walker, R. (2003, June 16) Dacey: Agencies need smarter, stronger security management. Government Computer News.

Walker, R. (2003, June 16) Dacey: As threats rise, feds shelter their IT. Government Computer News.

WebMethods. (2002, May) Demystifying Web Services.

WebMethods. (2002, May) Implementing Enterprise Web Services with the WebMethods Integration Platform.

WebMethods. (2002, May) Requirements for Securing Enterprise Web Services.

Zyskowski, J. (2003, June 2) No one said it would be easy. Federal Computer Week., p. S1.

A Success Story: Concord's Use of Web Services

I. Introduction

Concord EFS, Inc., headquartered in Memphis, TN, is a leading electronic commerce provider. With over 400,000 merchants collectively generating 10.8 billion transactions in 2002, Concord EFS is one of the largest electronic payment processors in the United States. Concord EFS specializes in fast, secure transaction processing at the point-of-sale (POS) and online by providing comprehensive single-source solutions for credit, debit, gift, and check services processing (www.concordefsnet.com). Employing more than 2,800 employees, Concord has 6,283 financial institution clients, and some 300,000 acquired merchant locations. With an $8+ billion market, Concord is listed on the S&P 500, NASDAQ 100, Fortune 1000, and Forbes 500. In 2001 it received the Wall Street Journal "Shareholder Scoreboard Honor Roll" and Forbes' Platinum 400 "Best Companies in America." Concord's payment services clients include, but are not limited to: Wal Mart, Piggly Wiggly, PETCO, Ryder, A&P, Rite Aid, Sunoco, BJ's, Wawa, Gulf, PetSmart, Sheetz, Vans, and Exxon Mobile. Some of their network services clients include SunTrust, Bank of Hawaii, Citibank, Bank One, Wachovia, Wells Fargo, American Express, Bank of America, and Sovereign Bank (Bricker, Roy).

This Case Study describes a problem that Concord was having with their electronic financial transaction processing, a successful solution that addresses the problem, and the benefits of the solution. It also discusses the significance of some of the outcomes and lessons learned from the implementation of the solution. Read on and learn how Concord discovered that web applications and databases alone were not enough to solve their problems. You will also see how this Web Services solution not only enhanced Concord's capabilities and performance, but also positively affected their vast customer base as well.

II. Statement of the Problem

In the area of electronic (Internet) payment processing, merchants' computer applications, code, computing platforms, and architectures are very diverse, and there existed within Concord a strong need for interoperability in and among the payment process components. There was a need for a common communication mechanism without necessitating all clients re-design and redevelop their network and computing environments. The single most difficult act to overcome was the integration of data from merchants' disparate platforms. Integration can be a long, arduous process, taking anywhere from three to six months (Lurie). A web site that is set up to accept payments must access a payment gateway that takes the payment data (credit card number, expiration date, etc.) and passes it on to a payment processor, which in turn is responsible for moving the money from one account to the other. Without standards, the integration affording the ability to communicate with proprietary systems was a nightmare.

Connecting to the gateway can take many forms: dial-up line, leased line, or even satellite. Traditional technologies relied on costly, complex, and proprietary telecom links (Salas, Jason). The POS market had not seen much advancement in communication technologies for several

years, and delays were typical with the slow line speeds and outages associated with dial-up and leased lines. Concord provided only closed-network access, and transactions over these lines took as long as 20 seconds each to process. Latency in the sales process is not tolerated in a competitive market. Vans (the worldwide Core Sports brand leader) had experienced delays in handling credit card payments due to the outages incurred when the manual off-line process was used. This frustrated customers. Time is a precious commodity, and once shoppers decide on purchases, whether on-line, or in a brick and mortar POS environment, they want to pay for the merchandise and continue on with the agenda of the day.

There was clearly a need for a solid, interoperable, secure, efficient, electronic payment system for online shopping, as well as brick-and-mortar, POS environments that would integrate disparate merchant platforms and employ a friendly interface.

III. The Solution

The browser environment is not the monoculture that it once was. It has progressed to providing Web Services that deliver robust capabilities in the area of information sharing/data integration. Conceptually, the term "Web Services" is a no-brainer (a service offered on the web), but technologically involves "component services" – self contained modular business applications for things like claims processing, currency conversion, and language translation that service providers such as web hosts and content delivery network operators can offer customers, who can use these modules to build their own Web Services (Tanner, 1 Feb 02). Concord EFS developed a Web Services Internet payment gateway entitled EFSnet. The gateway acts as a translator between Internet language and payment processing language to help dissimilar systems communicate (Krill). This solution safely integrated the data from customers' disparate processing environments.

As a Web Services solution, EFSnet uses the Internet as a transport mechanism, and integrates with most platforms and programming languages (Lurie). Standards enabled integration; EFSnet made the integration with merchant platforms possible by basing their development on a Web Services model, utilizing multiple platform-independent messaging protocols such as Simple Object Access Protocol (SOAP), Common Gateway Interface (CGI), and Extensible Markup Language (XML). XML allows companies to do business with each other using the web by bridging the differences between each company's individual proprietary network systems (Tanner, 19 Feb 03).

EFSnet did not ignore security. Data is encrypted using 128-bit Secure Socket Layers (SSL) (HTTPS). Firewalls, database encryption, and rules-based monitoring are also employed.

The EFSnet solution exists in two geographically distinct data centers in which the "Noah's Ark" approach is employed for redundancy.

While slow, dial-up transaction speeds were a major concern, the new three-second Internet transaction speeds solved this problem (Peters). Stores deploying the Web Services solution can

utilize any Internet connection such as a DSL, cable, or leased line connection and avoid the six-to-eight-second delay associated with dial-up connections when processing payments, according to Concord (www.concordefsnet.com).

Concord EFSnet costs the customers an initial setup fee of about $200, plus an annual fee of roughly the same price, with each gateway transaction processed costing a few cents each.

EFSnet Web Payment Services leverages the many benefits associated with Internet communication including low-cost, high-speed, high-scalability, global-accessibility, secure, interoperable, and multi-use capability. This diagram depicts the process flow of a transaction: (www.http://www.concordefsnet.com/Home/ProdEfsnetTheProcessor.asp):

IV. Benefits of the Solution

There are a number of inherent advantages associated with the EFSnet solution. The most important one is the ability for EFSnet to communicate securely and efficiently to all of their merchants' computing environments. By adhering to the Internet standards such as HTTPS, XML, SOAP, and CGI, development of EFSnet applications can be accomplished on disparate platforms using a wide variety of development languages and tools, without having to change or reconfigure existing merchant applications. EFSnet reduces the traditional barriers of programming by acting as a translator between standard Internet language and Concord's payment processing language. Where it used to take weeks and months to develop and implement payment solutions for traditional communication mediums, developing applications and devices to work with EFSnet can be accomplished within a timeframe of days and hours (www.concordefsnet.com). Integration costs are also decreased (Econworld.com, January 2002).

Transaction speeds are down from 8-20 seconds to 2-3 seconds. Dedicated lines are no longer needed; an Internet connection is all that is necessary. Fast Internet connections decreased costs and increased revenues. EFSnet experiences an uptime of 99.999% with less than 10% of its capacity utilized at any peak time. In the aforementioned Vans example, employment of the EFSnet solution resulted in efficiencies with the payment process in that it improved responsiveness to consumers by reducing their payment authorization time.

EFSnet can post enhancements without disruption in service and without having to port software to merchant locations. Changes to the application are seamless.

"It takes away the entire complexity of payments," said Roy Bricker, vice president of strategic alliances for Web payment services at Concord, in Tempe, AZ (Bricker, Roy).

V. Significance of Outcomes and Lessons Learned

The single most important benefit of Web Services technology is the ability to quickly and safely integrate payment operations from disparate platforms by dynamically exchanging data internally and externally. Merchants' application re-writing is not required; legacy applications may be reused.

However, by Concord bringing more merchants to market quicker, transactions processing revenue began sooner, and more transactions were processed. Providing Web services has been a win-win for Concord and its customers. EFSnet's successful use of web Services played a major part in the sale of Concord to First Data in April 2003. Concord's real-time payments expertise and capabilities provide customers with the most innovative and cost effective solutions for all of their electronic payment needs (Berniker, M.).

Concord operates with a relatively small number of employees, considering their customer base. This must be largely due to the Web Services technology economies of scale.

A huge lesson learned is that security is most critical with Web Services since this exposes an organization's internal operations. EFSnet not only saves time and money; it is secure!

VI. Conclusion

From restaurants and retailers, to supermarkets and online merchants, EFSnet provides a robust, low-cost, secure payment solution that takes full advantage of Web Services and data integration technology. Web Services technology represents a significant step in the evolution of distributed systems and promises to transform computing by providing universal interoperability between applications running on different platforms. Much energy and attention has been devoted to web Services and the development of standards such as SOAP. Standards bodies continue to address areas such as security to ensure that Web Services will continue to succeed.

References

Berniker, Mark. "First Data Buying Concord EFS for $7 Billion." April 2, 2002, retrieved June 15, 2003, from www.boston.internet.com/news/print.php/2173861

Bricker, Roy (VP Business Development & Alliances, Emerging Technologies Group). Concord EFSnet – Web Services Case Study, 2001.

Concord – The Payment Processor, retrieved 15 June 2003 from www.concordefsnet.com/Home/ProdEfsnetTheProcessor.asp

Krill, Paul. "Brick-and-mortars get credit card authorization via Internet." January 27, 2002, InfoWorld, www.infoworld.com/article/02/01/17/020117hnconcordefs_1.html

Lurie, Jonathan. "Web Services: Putting theory into practice." July 14, 2002 www.asia.cnet.com/builder/architect/web/printfriendly.htm…

Peters, Kurt. "Credit to the Web," Internet Retailer, November 2002 http://www.internetretailer.com/article.asp?id=7991.

Salas, Jason. "Accelerating Electronic Payment Processing," .net Magazine, retrieved 15 June 2003, from www.fawcette.com/dotnetmag/2002_05/magazine/columns/casestudies/concord/defa…

Tanner, John C. "Having it both ways: transactions on the edge," February 1, 2002, retrieved 15 June 2003 from http://www.telecmasia.net/telecomasia/article/articleDetail.jsp?id=12646

Tanner, John C. "Web Services: the five-year plan," February 1, 2003, retrieved 15 June 2003 from http://www.telecomasia.net/telecomasia/article/articleDetail.jsp?id=45860