Web Services Implementation:
The Beta Phase of EPA Network Nodes

Connie Dwyer and Chris Clark
U.S. Environmental Protection Agency, 1200 Pennsylvania Avenue, N. W., Washington, D.C.
dwyer.connie@epa.gov or clark.chris@epa.gov

Mark Nobles
Logistics Management Institute, 2000 Corporate Ridge, McLean VA 22102-7805
mnobles@lmi.org

ABSTRACT

This paper describes the Beta Phase of the Network Nodes pilot effort. The paper includes a brief introduction to XML and SOAP, a description of the mechanism used to generate nodes service requests, the general architecture of the State/EPA nodes, and the security measures used to protect the data exchanges. The Beta Phase concentrated on the development of Nodes that can respond to SOAP messages containing service requests (queries) against a State/EPA environmental database system. A query against a remote database via the Internet is often referred to as a web service. This pilot effort is the second phase of an initiative to institutionalize web services as a legitimate method for exchanging environmental data.

INTRODUCTION

The Network Blueprint1 postulates the creation of network nodes by the States and the U.S. Environmental Protection Agency (EPA) to exchange environmental data across the Internet. The Interim Network Steering Group (INSG) sponsored a cooperative effort by the States and EPA to demonstrate that the concepts of the Blueprint are reliable for day-to-day environmental data exchanges based on using eXtensible Markup Language (XML) and the Simple Object Access Protocol (SOAP). Six states (Delaware, Florida, Nebraska, New Hampshire, New Mexico, and Utah) and the EPA's Central Data Exchange (CDX) actively developed nodes that exchange environmental data seamlessly. The teams developing nodes used a variety of commercial software products to create network nodes that effectively respond to service requests (queries) submitted over the Internet. The pilot effort recently completed its second (Beta) Phase -- this document describes the approach and the results of those efforts.

BODY

The web services implemented during the Beta Phase were based on three service requests (queries) that returned facility data.

- Query 1 returned changes in facility data based upon an "As of Change Date" supplied by the requestor. The response to Query 1 contains nine data sets that conform to nine schemas developed by the Facility Data Action Team.

- Query 2 returned facility data based upon two user-supplied parameters: a facility name and/or an environmental interest. The response to Query 2 conforms to an abbreviated version of the consolidated schema used for Query 3.

- Query 3 allows the user to query for facility data based upon either a State ID or an FRS ID. The response to Query 3 conforms to a consolidated version of the nine facility data schema used to support Query 1.

The service requests exchanged between nodes were XML documents enveloped using SOAP. Since XML and SOAP are relatively new technologies and some readers may not be familiar with them, the next two sections briefly introduce XML and SOAP so that the reader will have a foundation for the remainder of the discussion.

XML

The eXtensible Markup Language (XML) is an outgrowth of SGML and provides a method for describing data based upon a standard syntax developed by the World Wide Web Consortium (W3C). XML is a vendor neutral approach for describing data. One popular use of XML is the exchange of data between database applications. One strength of XML is its ability to provide self-describing data -- the description is achieved with tags that provide the context for the data. For instance, an XML document may convey a name as:

```
<Name>
    <FirstName>Jane</FirstName>
    <LastName>Doe</LastName>
</Name>
```

The XML standards provide several approaches to building templates that describe the structure and format of XML documents. The Nodes Beta Phase used XML schema to describe the queries and responses that the nodes would accept. The group developed a schema for the service requests and modified schemas developed by the Facilities Data Action Team for the responses. For readers who wish to investigate XML further, the XML specification is available on the Internet at "http://www.w3.org/TR/xml11/".

SOAP

The Beta Phase used Simple Object Access Protocol (SOAP) as the enveloping structure for the Nodes service requests. Many software tool vendors (Microsoft, IBM, Oracle, SUN, etc) support SOAP in their product lines. SOAP has a binding to HTTP that allows for easy use across the Internet on standard ports that are typically open in firewalls (port 80 for HTTP and port 443 for HTTPS). Although SOAP can be used with other transport mechanisms (e.g. SMTP e-mail), the Beta Phase only used SOAP with HTTP and HTTPS.

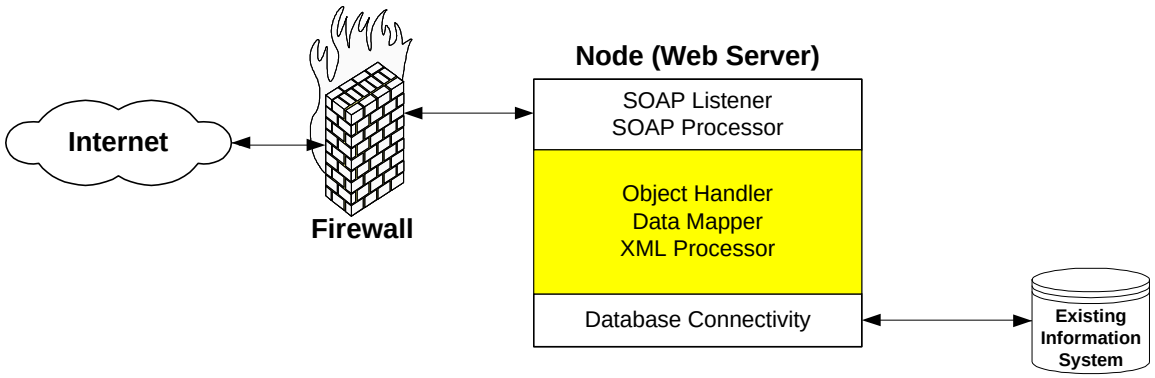
The SOAP specification defines an optional header section and a mandatory body section. The body section carries the nodes service request from the initiator and the response data from the servicing node. The SOAP body section may also carry error responses. For those interested in a more in-depth discussion of SOAP, the most recent version of the W3C SOAP specification is on the Internet at "http://www.w3.org/TR/soap12-part1/".

Node Architecture

A generic node accepts SOAP messages transmitted across the Internet using HTTP/HTTPS. In many cases, the node will be protected by a firewall. The web server associated with a node will interface with SOAP modules that interpret the SOAP message and hand the service request off for further processing by a middleware tier that invokes a query against the database. The Beta Phase nodes used a variety of middleware products (X-Aware, BizTalk, and components of the Oracle XML Development Kit). As the use of XML and SOAP increases, the requirement for middleware may be

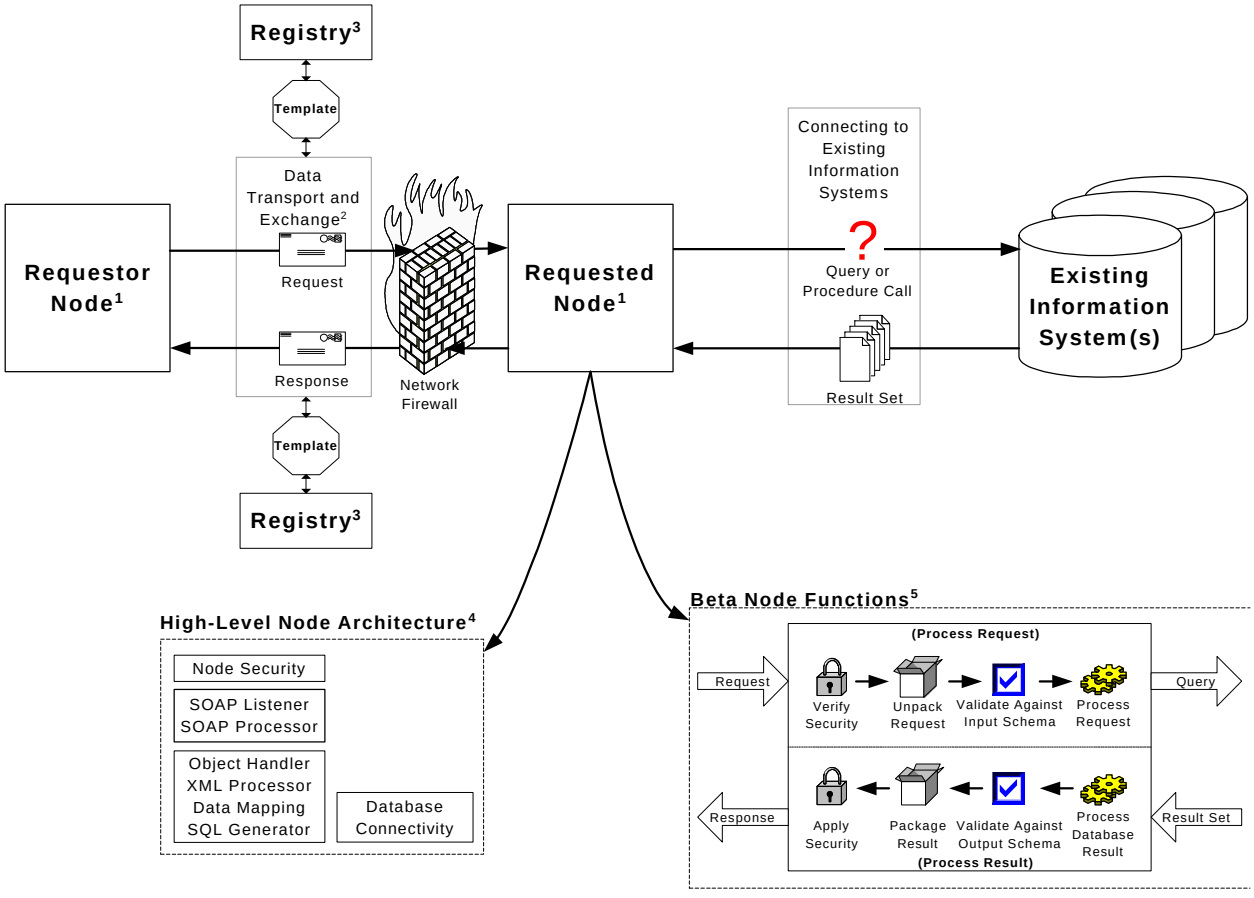
reduced as the major software vendors incorporate this middleware functionality into their mainline products such as application servers or web servers. Figure 1 depicts an overview of basic node architecture. This figure and all figures from this document are extracted from *Network Node Pilot Project Beta Phase: Report on Project Results and Next Steps*.

Figure 1. Basic Node Architecture



Within the Beta Phase pilot, the generic architecture evolved to that shown below in Figure 2. In this architecture, the nodes retrieved the schemas for the service requests and the responses from a repository. This ensured that all nodes used the same version of the schemas for validation. Use of a repository is a common theme in implementation of web services. The W3C is developing specifications that will facilitate searching repositories to determine the types of web services offered by an organization.

Figure 2. Beta Phase Node Architecture



The notes below explain the superscripts in Figure 2.

1. Requestor Node functions include making requests and processing responses from requests. Requested Node functions include processing incoming requests and processing results sets from existing information systems.

2. The Network Steering Board will develop exchange protocols and implementation standards. This will likely include WSDL.

3. DET (Template) validation could occur up to four times for each transaction. The beta Phase participants recommend that the Board (TRG) further investigate Template validation.

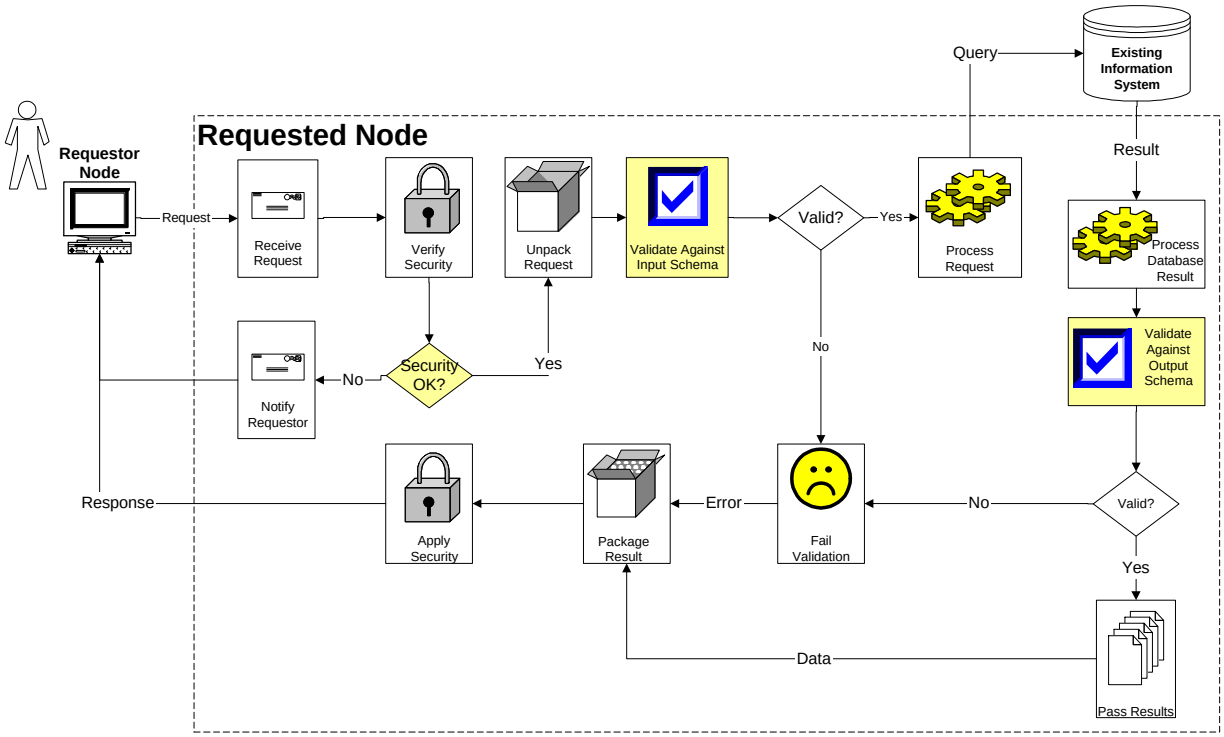
4. Technical Specifications for each Partner's Node will be based on total choices and their own computing environment. They will be developed by each Partner.

5. Functional specifications will be developed by the Network Steering Board and shared with all Partners.

Node Processing

A node receiving a web services request follows a well-defined series of steps to generate the response. Figure 3 illustrates these processing steps. Many of these steps are handled in a nearly automatic fashion. A good example is the security functionality. When the web server is configured for the SSL protocol, the encryption and decryption require no additional coding by node developers. The validation against schema is invoked by adding one additional parameter to the call that invokes the XML parser. The most significant step in the processing flow is the query against the database. Developing these queries took a significant amount of time. As existing nodes add new web services, the largest resource commitment may well be the time required by the database administrators to develop the queries for new service request.

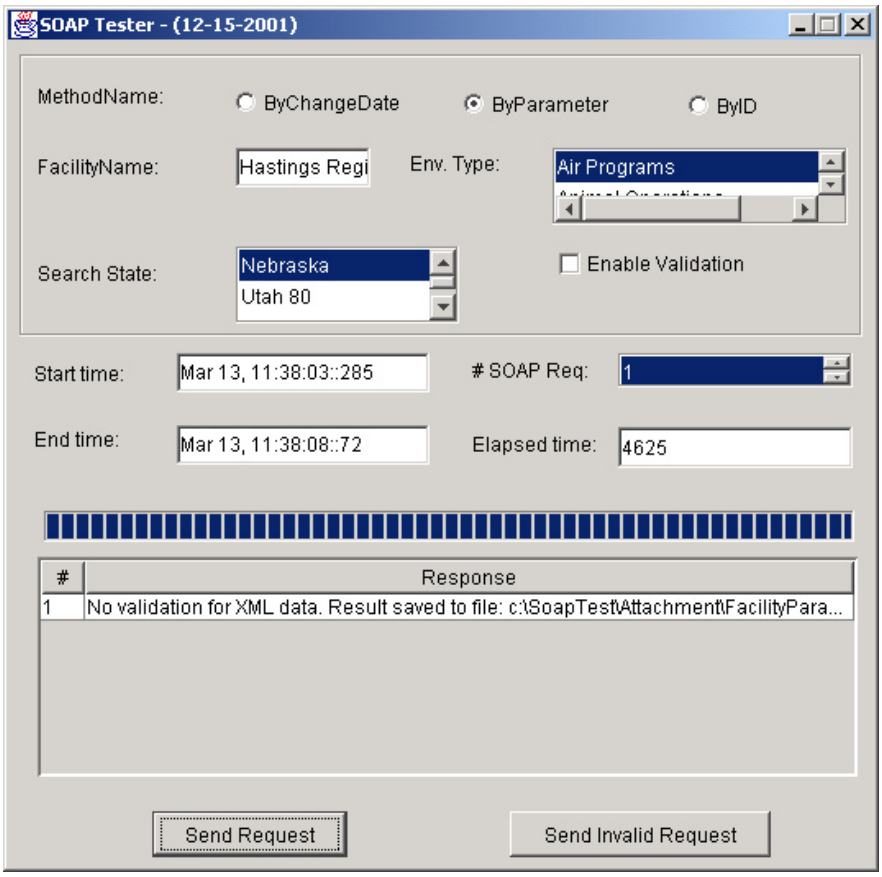
Figure 3. Processing Steps After Receipt of a Service Request



Java Test Application

During the Beta Phase, the emphasis was on a node's capability to respond to a service request. To enable easy generation of service requests, the Beta Phase relied upon a Java test application that provided a graphical user interface (GUI). The GUI allowed easy selection of one of the three queries and input of required parameters. The Java test application generated a service request that complied with the Nodes service request schema, wrapped the service request in a SOAP envelope, and transmitted the request to the user-selected node. The Java test application also processed the response, updated the GUI display with timing metrics, and stored the response on the user's hard drive. Figure 4 is a screen shot of the GUI display configured for Query 2, a query based on a facility name and environmental interest supplied by the user.

Figure 4. Nodes Service Request 2 Display



Although the use of the Java test application made generation of service request very easy, this probably will not be the model for day-to-day use. Since many of the node exchanges will be machine-to-machine, a GUI display for human interaction will not be as necessary. Tool vendors are also making the generation of request against web service easier. In many cases, wizards are available for reading files that describe web services and creating the code necessary to invoke the web service. As the sophistication of web services increase, the need for GUI interfaces should decline accept for testing purposes.

Security

The Network Blueprint defines categories of security that provide for secure exchange of data between network nodes. Table 1 lists the four levels of security defined in the Network Blueprint.

Table 1. Network Blueprint Security Levels

Security Level	Characteristics	Approach
Level 1	Public information that requires no authentication or certification of integrity. Like all Network information, this information is protected from unauthorized modification at its node.	This information will be available through the Internet on a public, non-secure website. Information can be transmitted without encryption or special security measures.
Level 2	Information that requires some additional level of authentication (i.e. that it is the State who is submitting the data) and a higher level of integrity protection. This data may require some level of confidentiality.	This information will be available through the Internet on a website that is secured using Secure Sockets Layer (SSL). The use of SSL allows the users to authenticate that the site being accessed is an approved environmental agency website, and provides privacy by encrypting all data in transit. SSL also provides data integrity protection.
Level 3	Information at this level requires bi-directional authentication and a higher level of confidentiality. All data submitted by users to environmental agencies is to be treated at this level or higher. This data is of a highly sensitive nature passed between agencies but does not require digital signature. This level can apply to person-to-person and server-to-server transactions.	Access to this information is protected by SSL at the server level, and by the requirement for user's digital identity credentials. These credentials will be in the form of X.509 version 3 digital certificates issued by a Public Key Infrastructure (PKI) that the environmental agency determines meets a sufficient level of assurance in identity proofing and credential protection. Once users have been authenticated, they will be permitted to access only that data to which they are allowed.
Level 4	Information protection that requires non-repudiation in addition to privacy, authentication and data integrity. Generally, this information is the electronic version of current paper processes that require an ink signature. This information may be in the form of data going from the agency to external users, or may be reports, applications or other information going from external users to the environmental agency.	This information will be protected by requiring a digital signature "affixed" to the data that can be validated at the time of acceptance of the information by the environmental agency or the external user. Digital certificates issued by an approved PKI will be used for the digital signature.

The Beta Phase of the network nodes pilot implemented Levels 1 and 2 of the four levels defined in the Network Blueprint. Each Node had two URLS: one for transmission "in the clear" using HTTP and another for confidential transmission using SSL via HTTPS. The EPA provided SSL server certificates for five of the state node participants so that the states could configure their servers for SSL.

The data encryption provided by SSL requires additional processing by the processors at either end of the network exchange. The participants in the Beta Phase used the Java Test Application to measure the additional time required for SSL processing. As expected, SSL did slightly increase the time required for the round trip transmission, but the increase was manageable. Table 2 from the Beta Phase final report summarizes the metrics associated with HTTP and HTTPS transmission of node data exchanges.

Table 2. Network Security Level 1 (port 80) vs. Network Security Level 2 (port 443)

	Average Time Port 80	Standard Deviation Port 80	Average Time Port 443	Standard Deviation Port 443
Test 1 (1300 records) 800KB *ByParameter* query	17.23 Seconds	± 1.66 Seconds	20.30 Seconds	± 2.61 Seconds
Test 2 (1 record) 10KB *ByID* query	1.11 Seconds	± .21 Seconds	1.57 Seconds	± .46 Seconds

A number of variables can affect the timing of round-trip transmissions across the Internet. Based on the findings of the Beta Phase, SSL does introduce small timing increases. These timing increases can be attributed to the additional processing required to encrypt and decrypt the data exchanges. As traffic increases between the nodes, some may view the SSL encryption as an unnecessary performance degradation. One mitigation approach would be the use of SSL accelerators that offload the encryption/decryption function to a supplementary processor. These hardware devices are readily available and could allow the continued use of SSL to provide confidentiality while increasing throughput.

Another option may be the use of XML compression to reduce the file size of the response before transmission across the Internet. The W3C is working on compression standards for XML documents but these standards were not complete enough for use during the Beta Phase. After these standards mature, XML compression may provide an offset for the SSL encryption overhead.

Mutually Authenticated SSL

Level 3 security requires both parties to authenticate to each other based on their X.509 certificates. The Beta Phase did not attempt mutually authenticated SSL because the Java Test Application that generated the service requests would have required additional modification to support the mutual authentication handshaking protocol. Since the States and EPA do not intend to use the Java Test Application in future iterations of the Network, the group elected not to modify the program.

Digital Signatures

The fourth security level requires digital signatures. The Beta Phase did not attempt Level 4 security because the XML digital signature standard was still evolving. Incorporation of digital signatures over the SOAP message will be attempted in later efforts to further define Node data exchange protocols.

LESSONS LEARNED

RPC

The W3C SOAP specification provides for two basic schema constructs: document-oriented and procedure-oriented (remote procedure calls = RPCs). The RPC approach places the method (function) name in the first element of the SOAP body section. The schema describing the nodes service request was written as a document centric schema, which places no restrictions on where the method name is located. During subsequent node development, the group learned that many of the software vendors have included specialized classes that make it very easy to deal with RPCs.

Since the service requests did not conform to the RPC approach, the developers had to parse the inbound service request to get to the method name before they could use the specialized RPC classes

provided by the software vendors. Although this was not difficult, it did create additional coding and did not take full advantage of the capability offered by the software vendors.

Many of the web services that the States and EPA will use are good candidates for RPCs. Developers of new environmental web services should consider the RPC model as the first option for describing new web services. Where the RPC model will work, RPCs should be used due to the inherent capability of the commercial software packages to deal most efficiently with this type of SOAP request.

WSDL

Web Services Description Language (WSDL) is one of the family of standards developed by the W3C to support web services. WSDL is an XML format for describing network services. The Beta Phase did not initially implement WSDL because WSDL was not widely supported by software tool vendors due to its immaturity. As part of their node development, the Florida team used a candidate release of Oracle's JDeveloper tools to create a WSDL file that described their web service. The EPA CDX node used the JDeveloper wizards to read the Florida WSDL file and automatically create Java code that could query the Florida node. This process took only a few hours and demonstrates the utility of WSDL as a method to make generation of queries against web services more efficient. As the use of WSDL becomes more common, organizations will use repositories to store links to their WSDL files and the schemas that they support. Users will be able to search for these using another emerging capability, Universal Description, Discovery and Integration (UDDI).

Non-Standard Ports

Some of the state node implementations used non-standard ports (high ports) for the HTTP/HTTPS services. Some firewalls and proxy servers block access to these non-standard ports. At least one user behind the EPA firewall was unable to invoke that state's environmental web services. Node implementers should use the standard ports associated with specific web protocols to avoid firewall restrictions. If non-standard ports must be used, the clients of web service may have to reconfigure their system to gain access to the web service.

Synchronous versus Asynchronous Messaging

The web services implemented during the Beta Phase were all synchronous processes. As the volume of service requests hitting each environmental node increases, there may be times when asynchronous processing is desirable to allow efficient use of computing resources. Some requests may require extensive processing to develop the response. increase the overall performance of the network, some nodes may delay processing of this type of request until a non-peak period. Any follow-on exchange protocol effort to the Beta phase should investigate mechanisms for incorporating asynchronous enablers into the schemas for the nodes service request. Asynchronous processing would also allow for responses being returned using other transport protocols such as SMTP e-mail or FTP.

CONCLUSIONS

The Beta Phase of the Network Nodes pilot has demonstrated conclusively that the States and EPA can use web services to exchange environmental data. These initial implementations also showed that network exchanges could be enhanced with emerging web services standards such as WSDL and UDDI. The final report on the Beta Phase of the Network Nodes Pilot contains a recommendation for another round of prototyping to further explore the use of these technologies and to develop and test Node Functional Specifications and Network Exchange Protocols. These preliminary steps will ensure that subsequent node development has a strong foundation on which to build.

REFERENCES

1. Blueprint for a National Environmental Information Exchange Network. http://www.sso.org/ecos/eie/COMPLETE_BLUEPRINT_JUNE_01_FINAL.pdf

2. Draft “Implementation Plan for the National Environmental Information Network”, January 29, 2002, Prepared for the Interim Network Steering Group by Ross and Associates Environmental Consulting, Ltd.

3. Draft “Network Node Pilot Project Beta Phase: Report on Project Results and Next Steps”, March 8, 2002, Prepared for the Network Steering Board by Ross and Associates Environmental Consulting, Ltd.

KEYWORD

Network Node

Web Services

SOAP

XML