Technical note— Web service credentials

by A. de Mes
E. Rongen

Web service technology promises advances in the dynamic integration of on-line services. Mechanisms are needed to assist in verifying the quality of these services. The end user of a Web service will quickly detect if the content is handled correctly and if the business process rules are adhered to, but often, many unwarranted assumptions are made concerning the quality of a service. Is the provider of the service reliable? If not, the consequences will primarily impact the user, who will blame the provider for delivering bad services. This paper explains the need for Web service credentials to address this issue and explores several aspects of an approach that facilitates a trustworthy dynamic integration of services.

Web service technology[1–3] allows new opportunities for inter-business collaboration in a very dynamic manner. On the lowest, technical level, Web service standards establish the definitions for the interface and binding properties of the service, Web Services Description Language (WSDL); the registry, Universal Description, Discovery, and Integration (UDDI), where the service definitions are published by the service provider and discovered by the service requestor; and the communication protocol, Simple Object Access Protocol (SOAP), to bind and invoke services (see Figure 1).

At a higher level, companies must agree on the terms on which they will do business. Today, a trading partner agreement (TPA) is negotiated "manually" between representatives of the two organizations, thus establishing mutual confidence between them. Several Web-service-related efforts are now focusing on defining the frameworks that describe how these TPAs may be established automatically, to allow businesses to engage in collaborations "on the fly."[4] Although some businesses may still have long-standing relationships (e.g. a customer and a preferred supplier), the collaboration between other businesses may exist only for the duration of one transaction.

As in conventional business, in this model of volatile collaborations, it is crucial that mutual trust exist between the two parties concerning matters such as quality of service. This goes beyond the issue of authentication, for which certification authorities such as VeriSign** provide solutions.[5]

Building confidence. Confidence in others is a prerequisite for all types of commerce. If you meet someone on the street who is selling watches, you will be hesitant to believe him when he indicates that the watches are solid gold Rolex** watches. Instead, when you want to buy a watch, you go to a merchant who inspires confidence. You seek independent references who provide evidence of the qualities that you are looking for. When in doubt about the reliability of others, you ask for input from their current and former customers.

0018-8670/03/$5.00 © 2003 IBM

For commerce to commence, your confidence level in a provider must be sufficient. In the context of dynamic integration, the same concerns prevail.

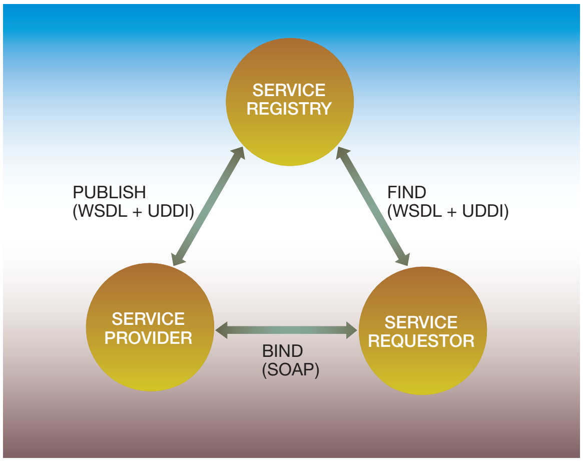
Consider the example of a requestor being a broker of services provided by other companies. In this scenario, an end user uses the application provided by the requestor to create a personal package of services that are provided by several businesses and brokered by the requestor. If a service of an unreliable service provider is exposed to a user, the user will complain to the requestor, and the user's customer satisfaction will be reduced. The requestor needs confidence about the provider prior to making the service available to users. This trust cannot be established by person-to-person negotiations for several reasons. The volatility and complexity of business collaborations makes human negotiations very labor-intensive, error-prone and time-consuming. For the duration of one transaction, multiple businesses may collaborate in a way they have never done before and will never do again. Therefore, new ways of establishing confidence between service requestor and service provider must be created. In the remainder of this technical note, we discuss in more detail how confidence can be built in this context, and how to check credentials using a mechanism based on the UDDI structure.

Currently, after manual negotiation of terms, the most widely used techniques to provide confidence between partners are:

- Authentication schemes established by third party certificate issuers (such as the Public Key Infrastructure, PKI). This ensures that parties know with whom they are dealing, but leaves them in doubt concerning reliability.
- Private trading networks, where an authority validates and taxes new traders and assures service quality. As members of the network, providers and requestors have a common place to go to in case of dispute. A drawback is that in order to become a member one may need to pay a membership fee or meet requirements which may be too resource-consuming.

Existing markets are complex, and transactions are often facilitated by brokers. This reduces freedom and openness in the selection of trading partners. In many cases, this broker model is unnecessarily limiting and can be improved by use of the solution outlined in this technical note.

Figure 1 Web services model



When interacting with a new partner, a third party can provide credentials to establish confidence. These credentials relate not only to the party's identity, but also to his or her ability to support all or part of the required business processes. In this context, three levels of confidence can be identified:

1. Confidence in a party's identity (authentication). A new partner will need to establish who he is. When identity is authenticated, the requestor can obtain the correct credential information.

2. Confidence in party's performance. Many authorities exist that measure and administer organizational performance, including:

   a) International standards bodies, for example, ISO (International Organization for Standardization) and SEI (Software Engineering Institute)
   b) National authorities, for example, BOVAG (Dutch Federation of Entrepreneurs in the Mobility Sector)
   c) Government authorities such as the military or the United Nations

3. General confidence in the trading party. This can be obtained through a consensus of opinions of other customers regarding the partner. These opinions may express satisfaction or dissatisfaction, make positive or negative recommendations, or both.

Figure 2 Description of a credential as a UDDI structure

```
<credential type="formal" name="ISO9000">
 <business uri="...uri to uddi entry"/>
 <authority uri="...uri to uddi entry"/>
 <issue date="2002-12-31"/>
 <expiry date="2004-01-15"/>
 <valid/>
</credential>
```

How individual credentials are valued to determine the level of confidence is a business decision and depends on the context in which they are used. In our model and in the remainder of this technical note, we will focus on measurable formal information provided by credentials. Opinions of customers and nonprovable statements by a provider can be introduced into the model, but this is not addressed here.

Credential model

In a UDDI registry, businesses register the Web services that they are offering. The credential model fits into the structure used in UDDI. See Figure 2 for an example of XML code describing a credential.

Before a business is permitted to add services into a UDDI registry, it must identify itself and provide details it considers relevant for business partners. In our model, this process is extended to include publication and verification of credentials. A business is required to identify itself in an indisputable manner. A reliable third party should confirm the business's identity. Identity needs confirmation in a legally acceptable form, for example, authentication through VeriSign, banks, or governmental agencies. Following this, the existing UDDI security mechanism suffices.

After identity is established, the business may start registering Web services. Independent of the Web services registered, the business may submit credentials in a standardized format.

Credentials (e.g., an ISO 9000 certificate) result from assessments and measurements by independent authorities that are recognized as qualified to perform them. The credential that a business submits contains information on this certification. Information on type of certification, date granted, expiry date, and issuing authority are required. Further information specific to the certificate can also be added. Several types of credentials are predefined, and the mechanism allows easy extension of credentials. Both the exchange and storage of the credential data should be in an XML-defined language.

Because the businesses themselves provide these credentials, the credentials need to be confirmed by the issuing authority (or their representative). A service in this model handles verification of the credentials (see Figure 3). This service resembles the UDDI service, and can be made equally centralized. Credentials are submitted by the provider and checked asynchronously until confirmed. To check a credential, this service contacts the issuing authority and logs confirmed credentials. In order to support these checks, issuing authorities should implement Web services that accept a credential description and return a validity statement and optional additional information, such as an expiry date.

Potential service requestors search for a service based on the classic search criteria (for example, the specific WSDL definition) and the credentials. The combination of credentials that gives a requestor enough confidence to interact with the provider could be defined in a formula, though making this 'formula for trust' public could work adversely for the requestor in some cases.

To check credentials, the requestor can submit an agent that will be run by the central UDDI server, or possibly on other hardware.[6] The requestor has the option, instead of providing source code for this agent, to supply a compiled module that applies (but does not publicize) its confidence formula. Adhering to the agent-naming trend (for example, "aglet"[7]), we call this agent a *credlet*. The credlet performs the search and applies the confidence formula. Credlets contain application code that may interact with their originating system and with the UDDI server.

To understand how the UDDI server interacts with the credlet, see Figure 4, which shows the steps that are used in discovering and using a credible service by means of credlets.

As shown in Figure 5, the initial set of services results from the UDDI query tag. The requestor can either specify a published XML rule to evaluate the services using the <rule> tag, or it can invoke an

Figure 3 Use of credentials and credlets in Web services

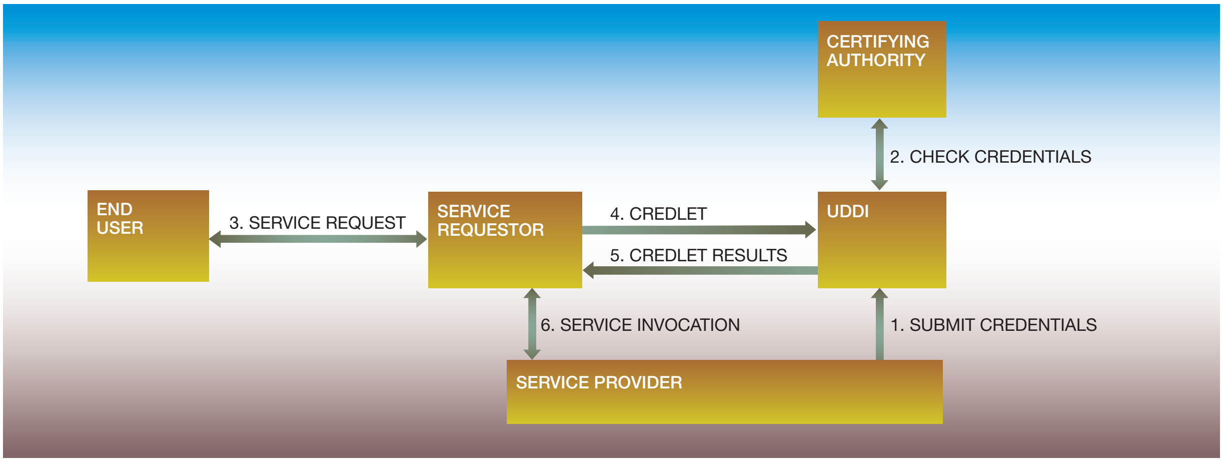


Table 1 Static versus dynamic binding of services

	Static Binding	Dynamic Binding
At design time	1. Invoke UDDI request for WSDL to create service invocation code. 2. Send credlet to UDDI.	
At run time	3. Invoke services.	1. Invoke UDDI request for WSDL to create service invocation code. 2. Send credlet to UDDI. 3. Invoke services.

evaluation service on a private server, specified by the <evaluationURL> tag.

One of the key advantages of Web services is the decoupling of implementation and transport protocol. An implementation-independent description of credlets is preferred. See Figure 5 for a solution of this type that uses XML.

Working with the model

Just as with the invocation of a target Web service, the requestor can obtain the information on the binding of a service at design time (static binding) or at runtime (dynamic binding).

It is likely that the selection and invocation of the Web services will be either both static (at design time) or both dynamic (at run time). As shown in Table 1, in the case of static binding, the credentials of the provider can be checked before the requestor application becomes operational, so that the requestor does not have to deal with the provider's credentials at run time. In many cases, the confidence-level check will be performed outside of the scope of the application, often as a manual task performed by employees of the requestor organization.

In the case of dynamic binding, the requestor can only check the credentials of the provider at run time because the providers whose services are going to be invoked are not known at design time. In this case, performing the confidence-level check manually is far too slow, and an automated check is preferable. To this end, the requestor can submit the credlet that will apply the confidence-level criteria after the initial set of providers is obtained from the UDDI. For each provider, the requestor decides whether to invoke the service based on the results of the evaluation of the rules.

Figure 4 Interaction of the UDDI server with a credlet

```
public interface Credlet {

        /**
        * Set initial values, and for example, go
        * back to the company server for private
        * information.
        */
        public void init();

        /**
        * Provide the query to search the UDDI for a
        * list of services that has to be evaluated.
        */
        public java.lang.String getUddiQuery();

        /**
        * This method is called for each Web service
        * that has to be evaluated. The evaluation is
        * either implemented in the credlet, or via calls
        * back to the company's home server for a
        * confidential evaluation. The services that
        * pass the test are added to the list of
        * credible services.
        */
        public void addService(Object service);

        /**
        * This method is called by the requestor who
        * initiated the credlet. listServices returns
        * the credible services.
        */
        public java.util.Enumeration listServices();
}
```

As an alternative to always checking the credentials, (for example, for performance reasons), the requestor can perform provider selection and credential checking on a regular basis (for example, once a day) instead of before each user request.

Conclusion

This technical note presents an approach in establishing trust and confidence between buyers and suppliers in a dynamic, on-line e-business scenario. Finding and selecting providers of a service is improved by adding credential checking to the process. Although we present the mechanism as an add-on to Web services, there is no hard link binding the mechanism to SOAP, WSDL, or UDDI; therefore, this mechanism may be applied in other business-to-business environments as well.

**Trademark or registered trademark of Rolex Watch U.S.A., Inc. or VeriSign, Inc.

Figure 5 Implementation-independent description of credlets

```
<credlet>
          <uddiquery>
          ....
          </uddiquery>
<!--rules for evaluating credentials-->
<!--specified by rule tag------------->
          <rule>
             <and>
                   <name="ISO9000"/>
                   <expired="false"/>
             </and>
          </rule>

<!--Alternatively, to keep rules------>
<!--private, one may specify a URL---->
<!--containing evaluation rules------->
<!--using the evaluationURL tag------->
<!--e.g.<evaluationURL=-->
<!--"http://www.myco.com/rules"/>-->
</credlet>
```

Cited references

1. *UDDI Technical White Paper* (September 6, 2000). Available at http://uddi.org.
2. D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer, "Simple Object Access Protocol (SOAP) 1.1," *World Wide Web Consortium* (May 8, 2000).
3. E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," *World Wide Web Consortium* (March 15, 2001).
4. A. Dan, D. M. Dias, R. Kearney, T. C. Lau, T. N. Nguyen, F. N. Parr, M. W. Sachs, and H. H. Shaikh, "Business-to-business Integration with tpaML and a Business-to-business Protocol Framework," *IBM Systems Journal* **40**, No. 1, 68–90 (2001).
5. *VeriSign Digital Trust Services: Enabling Trusted Web Services*, Verisign Corporation (February 12, 2002).
6. F. M. T. Brazier, C. M. Jonker, and J. Treur, "Compositional Design and Reuse of a Generic Agent Model," *Applied Artificial Intelligence Journal* **14**, No. 5, 491–538 (2000).
7. P. E. Clements, T. Papaioannou, and J. Edwards, "Aglets: Enabling the Virtual Enterprise," *Proceedings of the International Conference on Managing Enterprises—Stakeholders, Engineering, Logistics and Achievement*, (*ME-SELA 97*), Loughborough, UK (1997), pp. 425–431.

Accepted for publication February 18, 2003.

Arjan de Mes *IBM Global Services, P.O. Box 9999, 1006 CE Amsterdam, The Netherlands (mes@nl.ibm.com).* Mr. De Mes is an application architect and application development trend watcher in the Application Management Services organization of IBM Global Services. He joined IBM in 1995 as an application developer. In 1997 he started work as an application and infrastructure architect, working for customers in banking and finance as well as for IBM's internal account. In 2001 he joined the Architecture and Technology Innovation group where he had the opportunity to study, test and evaluate several recent technological advancements. His current interests include: service-oriented architecture, application performance, and the future of application development. Mr. De Mes received his M.Sc. degree in computer science from the University of Amsterdam in 1994.

Erik Rongen *IBM Global Services, P.O. Box 9999, 1006 CE Amsterdam, The Netherlands (erik@nl.ibm.com).* Dr. Rongen joined IBM in 1995 and since then has worked on various projects in the fields of e-business application development and the adaptation of legacy systems. He is interested in the impact of new technologies on software systems and the application development process. Currently, he monitors trends in application development for the Application Management Services organization and assesses emerging technologies for use in knowledge management systems. His interests include unstructured information management and the application of machine intelligence to custom application development. Dr. Rongen received his Ph.D. degree in physics at Delft University of Technology in the Netherlands.