Zeitschrift: Comtec : Informations- und Telekommunikationstechnologie = information and telecommunication technology

Herausgeber: Swisscom

Band: 79 (2001)

Heft: 7-8

Artikel: Web services : a new paradigm in software architecture

Autor: Messmer, Bruno / Keller, Rico / Clavadetscher, Charles

DOI: https://doi.org/10.5169/seals-876557

Exploration Programmes:
Corporate Technology Explores Future Telecommunications

Web Services: A New Paradigm in Software Architecture

Between April 8th and 11th, 2001, New York hosted the International XML Developers' Conference, where speakers of large enterprises agreed that the new paradigm of building web-accessible software components and publishing these components as Web Services will govern software development in the next five years. The Gartner Group, on the other hand, estimates that by the year 2006, 50% of midsize and large businesses will rely on Web Services. Judging from various trends, announcements and studies, it appears that a convergence is finally taking place towards a unified model of software architecture in the form of Web Services.

The Exploration Programme "The Net-Centric Application Business" explores the opportunity for remote applications and application service providing models that results from the expected availability of broadband Internet access, both fixed and mobile, and the evolution of various end-devices for residential and business customers.
With its Exploration Programmes, Corporate Technology is exploring telecommunication technologies and new service possibilities with a long-term view of 2–5 years. Further, the expertise built up in the course of this activity enables active support of business innovation projects.

BRUNO MESSMER, RICO KELLER AND CHARLES CLAVADETSCHER

In recent years, new trends and initiatives in software technology have been emerging on a monthly basis and have tried to capture the attention of the industry with headline announcements. Only few such trends have yet proven to be of substantial importance and even fewer deserve, in our opinion, to be called a new paradigm.
In its original Greek meaning, "paradigm" denotes the set of grammatical structures related to the flexion of words. In 1962, the science historian and philosopher Thomas Kuhn (1922–1996) reintroduced the term as related to the dialectical development of scientific disciplines. He defined a paradigm as the set of theories, laws, rules, models, concepts and definitions that go into a generally accepted fundamental theory of science. More interesting than a paradigm in itself is the way in which a new paradigm supersedes an existing one. Most philosophers agree that the emergence of a new paradigm, and the acceptance of it as such, means to deeply modify the way of viewing reality. It is a change in consciousness, i.e. in the thoughts and perceptions that form the intellectual foundation of a community. The point is that the subject of investigation remains the same, but what changes is the way people think about it.

It is with the promise of Web Services initiatives such as .NET that the Internet will be given a new functional depth as well as create new business opportunities. The openness and extensibility of the Web Services' interface will have a major impact on the development and the business models in various areas such as B2B marketplaces, Instant Messaging, Unified Communication, Application Service Provisioning and others.

```
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">
    <SOAP-ENV:Body>
        <m:Sum xmlns:m="Math-URI">
            <xValue>23.1</xValue>
            <yValue>11.4</yValue>
        </m:Sum>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">
    <SOAP-ENV:Body>
        <m:SumResponse xmlns:m="Math-URI">
            <sumResult>34.5</sumResult>
        </m:SumResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig. 1. A SOAP request as well as the response to it is a simple XML file that follows a specific Document Type Definition (DTD).

Web Services [1], or web-accessible software components, or, in even more general terms, software as a service, represent such a paradigm, which brings on new ways of thinking about how to solve the old problem of building software applications faster, more flexible and by aggregating components from different sources. Furthermore, Web Services bear the potential to bring a new dimension of functionality to the Internet and as such create new business opportunities. In short, the promises of Web Services are as follows:
- Interfaces to software services will be published and accessible via catalogue service.
- The coupling of components is compliant with open Internet standards.
- New applications can be built by aggregating and reusing these web-accessible software components.
- Software components can be sold as a service.
- Existing functionalities can be wrapped and sold as a service.

In particular, the last two points are of great interest to providers in the ASP arena and to companies like Swisscom, where the forming of new independent sub-companies leads to a repositioning in the value-chain and to questions on how functions can be provisioned in between the sub-companies.

Work at Corporate Technology

There are several questions raised by the opportunity for remote applications and application service providing models for net-centric applications. First, there is the question of how applications can be built in a distributed environment for a heterogeneous audience ranging from machines to humans with various handheld devices. Secondly, the management of customer data and customer relations will become a key competence for application service providers. Finally, it will not be sufficient to extend existing applications with Web or WAP interfaces in order to be successful, but new business models as well as new service ideas will be required in order to cash in on the said opportunity. It is one of the objectives of the exploration programme "The Net-Centric Application Business" to understand, assess and master the technology behind the potential for flexibility and hence adoption of services and applications based on Web Services.

The Emergence of Web Services

In the past, a developer implemented software keeping in mind that his program would run on a single computer with a single user and with a set of defined resources. The shift to the Client/Server paradigm was a very important step towards distributing computing time onto different computers. Nevertheless, as usual in first steps, software tended to be quite inflexible, defining proprietary protocols and interfaces each time. Additionally, software was developed for specific platforms which could hardly inter-operate. The Internet protocols and the definition of the Common Gateway Interface (CGI) helped overcome some of the difficulties, reducing network communication to the exchange of text. The upcoming of Java in the 90's and its enthusiastic acceptance in the developers' community freed software from being bound to a specific operating system. Developers soon perceived that writing software often implied re-inventing the wheel for each new product. The idea of components was born. Components are pieces of software that perform some defined operation and expose a set of functions for re-use in bigger applications. This allowed the creation of applications based on building blocks, reducing error proneness. As long as a single machine hosts all components from an application no great problems arise. But in the world of E-Business, companies are increasingly under pressure to link up their internal systems and exchange data with other companies in a dynamic and flexible manner so that heavy-weight protocols and technologies like EDI or CORBA are no longer feasible.

Spawned by these interests, there have been exciting new developments in the last two years. XML has been commonly recognised as a standard language for data formatting. Due to its flexible nature, XML is an ideal candidate for a new generation of communication protocols that connect up distributed software components past firewalls and over the public Internet. The most important effort in this direction was XML-RPC. XML-RPC allowed a method call to be done remotely by sending an XML file and retrieving the response as an XML file. Being designed for a specific purpose, however, XML-RPC lacked extensibility and flexibility. The industry was looking for something more advanced, which could encompass both remote procedure calls and data encoding at once. Based on the common ground of XML, large players in the software business such as Microsoft, SUN or HP have launched their own framework initiatives on how to build open Web Services [2]. Currently, the most prominent initiative is Microsoft's .NET framework. .NET combines a set of technologies and services in order to create a new generation of Internet services. The framework includes components such as C# (C-Sharp), a Java-like programming language, SOAP, a network protocol which enables these services to communicate over the Internet, UDDI, a directory service specification and also some central services such as Microsoft passport, a service which offers user identification and of course Hotmail.

While Microsoft clearly aims at dominating areas such as unified communication by integrating their central services with the new operating system Windows XP (and, hence, from a business point of view, poses a threat to incumbent operators), there is also a purely technical dimension of the .Net approach which contains some very interesting aspects with regard to Web Services architectures.

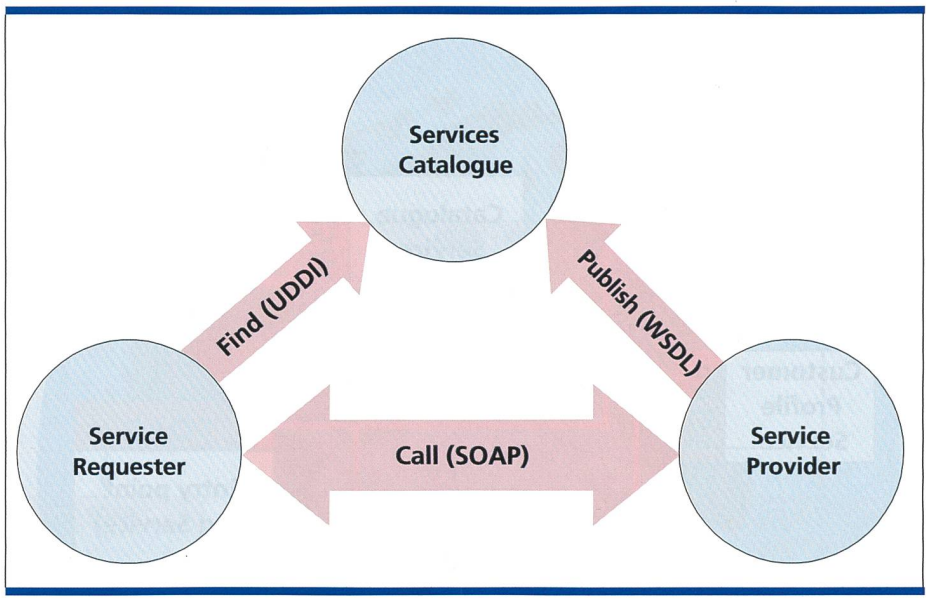

Fig. 2. Web Service Discovery and Invocation: Through a standardised description of the services provided, Web Services can be published and found easily on the network.

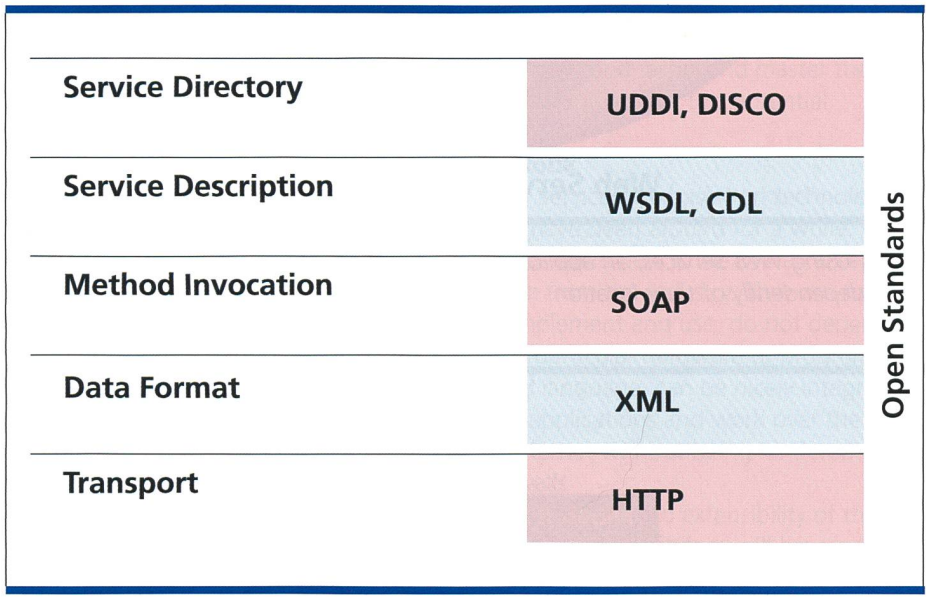

Fig. 3. The Web Service concept is based on open standards.

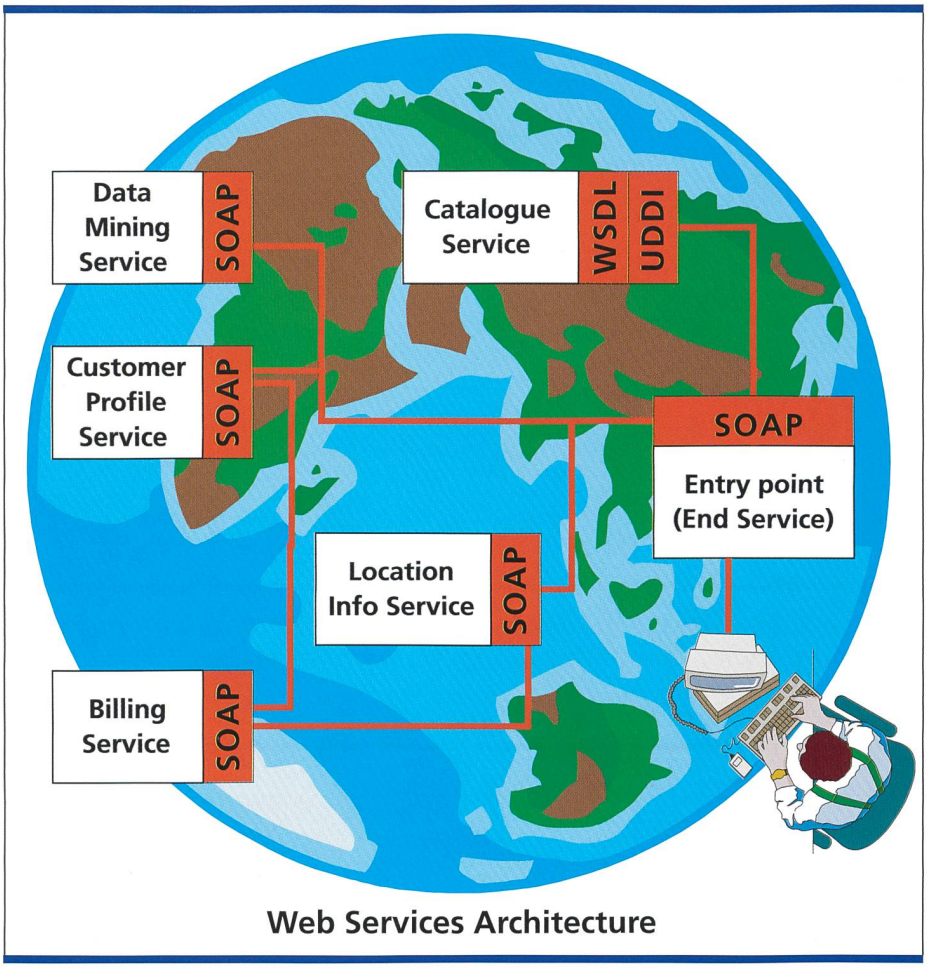

Fig. 4. Using Web Services, an application is a set of co-operating services that interact independently of their location.

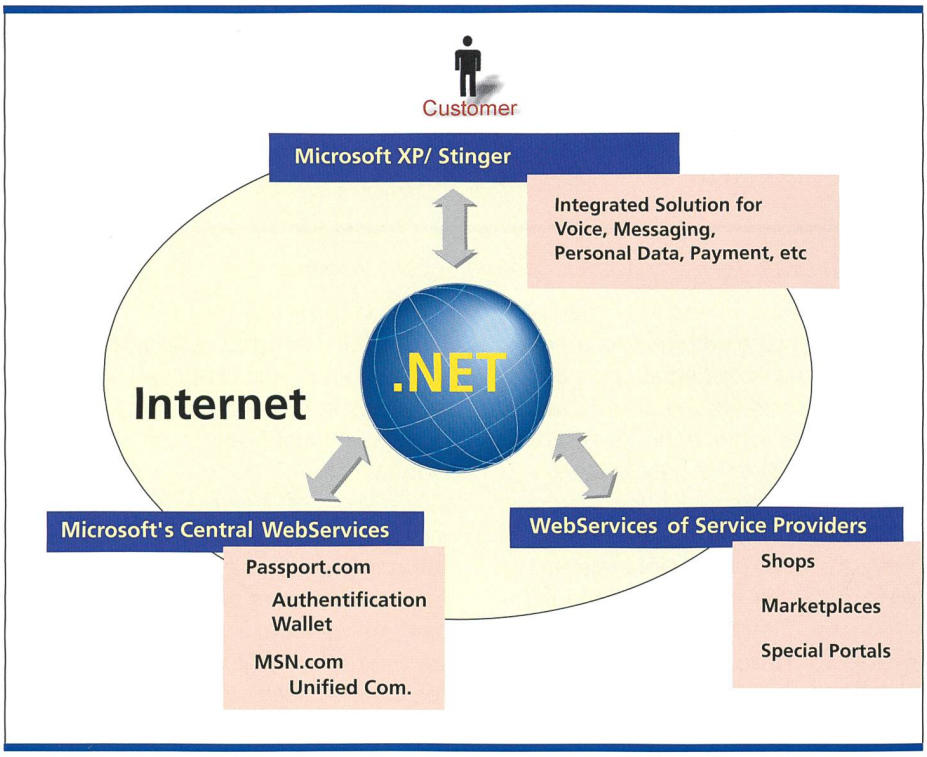

Fig. 5. .NET is Microsoft's platform for the development and integration of Web Services based applications.

The most interesting of these aspect is the remote method invocation protocol SOAP (Simple Object Access Protocol – or as IBM refers to it the Services-Oriented Architecture Protocol) which is a specification for making remote calls to functions by encoding the call and the parameters in an XML file which can then be sent over the network through HTTP. Being based on XML, it does not require cryptical encoding like CORBA or RMI. A SOAP envelope can be read and understood by a human being, thus allowing quick troubleshooting even with simple tools. Figure 1 shows a sample procedure call over SOAP and its response. The operation performed is a simple addition of two numbers with the name of the method to be called being defined in the body of the XML envelope. SOAP does not really define which transport protocol should be used. As a matter of fact, SOAP can be used over asynchronous systems like email. SOAP was designed and specified keeping simplicity in mind. For that reason many issues are not considered as a part of the specification, but as additional features that can be achieved on top of SOAP. A typical case is security in terms of authorisation as well as encryption.

SOAP has a big potential of being accepted as an industry standard, mainly due to its simplicity and openness. What is needed to get method calls working with SOAP is a XML parser and a SOAP enabled engine, both being components that are also available from open source projects such as Apache-SOAP.

However, while this is sufficient to get method calls working in a static environment where it is known beforehand what methods are available, where they are located, and what they do, the real power of the Web Service concept is only unleashed when methods and services can be found dynamically. For this purpose, the functionality, parameters, and result of the Web Service need to be described in a common language, and a mechanism for listing and finding the services is required. One emerging standard for the former is WSDL, the Web Service Description Language, which itself is based on XML. The problem of finding the Web Services is then approached by protocols such as UDDI, the Universal Discovery, Description and Integration specification [3]. UDDI is basically a service which can be used by service providers to publish Web Services

and by a service requester to find them. The common language for publishing and finding services is WSDL. Once a service requester has found the needed Web Service, it establishes a direct connection with this service. The communication between requester and provider is then performed over SOAP (fig. 2). In figure 3, an overview over the set of protocol and services layers involved in a Web Service concept is displayed. Note that, while on the transport and the method invocation layer the industry is commonly agreeing on standards like SOAP, the service description and the service discovery layers still lack the general support for a single protocol or specification.

However, it is the promise of initiatives such as .NET that the Internet will be given a new functional depth, as well as create new business opportunities. The functional depth consists of the fact that Web Services can be provided anywhere in the Internet and then aggregated into new services which can again be provided. The business opportunities arise due to the fact that there are new roles such as Web Services catalogue provider, E-services marketplaces, etc. Figure 4 illustrates how the different roles could interact independently of their location.

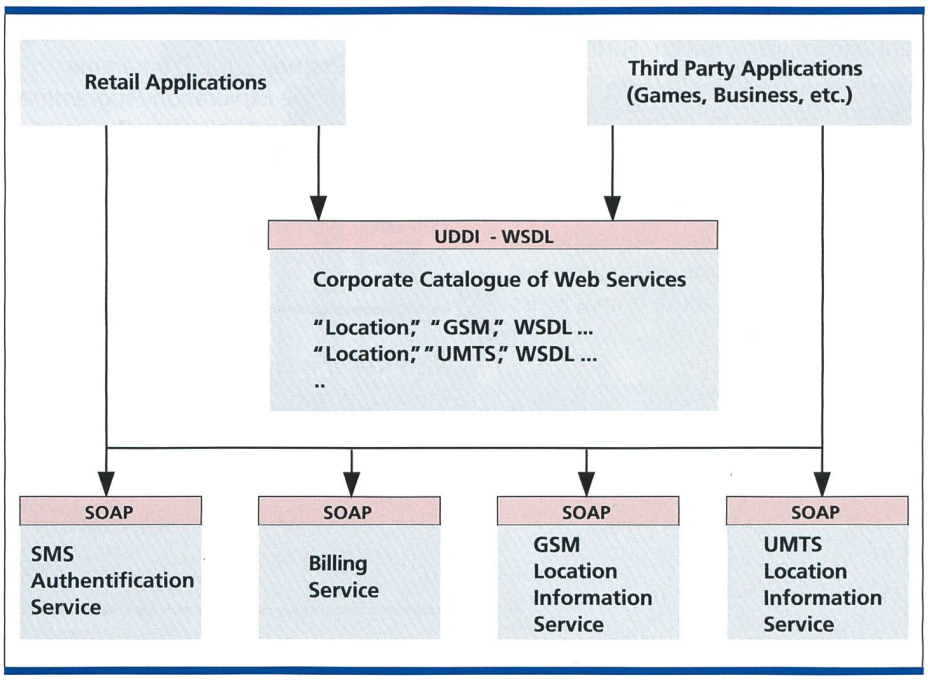

Fig. 6. Open, extensible and clear interfaces based on open standards allow great flexibility.

The Impact of the Web Services Initiatives

From the viewpoint of a telecom provider such as Swisscom there are two sides to the story of Web Services, and particularly, to the Microsoft .NET initiative. From a business point of view, the integration of Microsoft .NET services on the user's Windows XP desktop in combination with voice and identification services [4] may threaten some areas of the traditional telecommunication business (fig. 5). However, for Microsoft to become a full service provider, it is yet to be proven that crucial aspects such as security, privacy and reliability can be delivered.

The other side of the story is less related to a specific company's initiative but more to the underlying technology and the opportunities that arise for the process of building services. One particularly interesting aspect of the Web Services concept is that services can be dynamically aggregated into new applications and across company firewalls. In order to illustrate the poser of this concept, a situation of a mobile telecom operator is illustrated in figure 6 where the operator has different functionalities such as billing, authentification or location information which can be used to create a new kind of application. In the traditional approach, these functions would be provided as CORBA or RMI interfaces which could then be called (within firewalls) from other applications. The integration of third party providers, however, already poses a problem as firewalls must be passed and stub classes for connection must be provided. Here, Web Services offer an elegant solution, which also adds to the flexibility of applications. For example, all of the above mentioned functions could be provided via a SOAP interface and published in a corporate web services catalogue along with their WSDL description. Internal as well as external applications could then, instead of linking a CORBA call directly, search for a SOAP service "location" in the catalogue and call that service dynamically. Additionally, if at some point in time there is not only a service for "GSM location" but also for "UMTS location", the catalogue server could relay a request to the new service without the need for changes in the end applications. The potential for flexibility and hence adoption of services and applications based on Web Services is therefore large, and it is one of the objectives of the exploration programme "The Net-Centric Application Business" (see box) to understand, asses and master the technology behind this potential.

Outlook

Web Services are based on technologies that have been around for a while. What makes them new is the overall acceptance from the industry. They are simple to implement and use, do not depend on a particular hardware or programming language, can be nicely integrated into applications and work over the public Internet without being hindered by firewalls.

The openness and extensibility of the Web Services' interface will have a major impact on the development and the business models in various areas. Examples include

- Business-2-Business marketplaces
- Instant Messaging and Unified Communication [1]
- Application Service Provisioning
- Peer-2-Peer computing

According to the Gartner Group, by the year 2006, 50% of midsize and large businesses will rely on Web services. Hence, there is a large potential for various business roles to be filled in. Furthermore, many aspects of the Web Services concepts are not yet sufficiently specified or supported by technology. Mature products need to be able to handle transactions in a distributed environment. Having information running back and forth on the network it is absolutely

mandatory to have a clear security concept and implementation. Furthermore, the general architecture does not yet follow a standard and there is a remaining risk that firms might go different ways in spite of the unifying force of Web Services. Nonetheless, the Web Services vision has some very appealing aspects and Corporate Technology will continue to look at these from a Swisscom perspective. [10]

Bruno Messmer *is the Programme Manager of the Exploration Programme "The Net-Centric Application Business" of Swisscom Corporate Technology. He holds a PhD in Computer Science and has been working in various Internet and Software Projects.*

Rico Keller *is Project Leader in the Information Technology and Application domain at Swisscom Corporate Technology within the Exploration Programme "The Network Centric Application Business". He is responsible for the exploration in the area of e-market places.*

Charles Clavadetscher *works as a Research Engineer at Swisscom Corporate Technology and is responsible for the exploration of basic software infrastructure for emerging technologies as a project leader within "The Net-Centric Application Business" programme.*

References

[1] http://www4.gartner.com/5_about/press_room/pr20010501a.html
[2] http://www.webservices.org
[3] http://www.uddi.org
[4] http://www.passport.com

Abbreviations

API	Application Programming Interface. A defined set of function or method calls that can be used for interacting with a piece of software.
ASP	Application Service Provider
B2B	Business-2-Business
CORBA	Common Object Request Broker Architecture. A model for distributing computing ability on a network.
DTD	Document Type Definition. A file establishing a set of constraints for a XML file. DTD escapes the basic description style of XML and is possibly going to be substituted by XMLSchema.
EDI	Electronic Data Interchange
HTTP	Hypertext Transfer Protocol. The Internet standard for requesting and retrieving files in the WWW
JVM	Java Virtual Machine. A programme able to run compiled Java code. Java code is compiled in a so-called bytecode thus accelerating execution. In that sense Java is an interpreted but not a scripting language.
RMI	Remote Method Invocation. A Java implementation of the CORBA model. The use of RMI implies the existence of a JVM on both ends of the communication channel.
RPC	Remote Procedure Call. A function call made from one computer of a function residing on another computer.
SOAP	Simple Object Access Protocol. A proposed standard for RPC using XML over HTTP and for serialising and deserialising data.
UDDI	Universal Discovery, Description and Integration. A directory service allowing the registration and query of web services.
WSDL	Web Service Description Language. An XML file describing the way a web service can be accessed. I.e. which methods are exposed, which parameters are required, using which encoding and what return values may be expected.
WWW	World Wide Web. A generic information service platform. Often Internet and WWW are used as synonyms. Through the introduction of interactive pages the WWW is increasingly becoming an application platform.
XML	Extensible Markup Language. A new standard allowing everybody to define his own set of markup tags, i.e. formatting delimiters enclosed in brackets and following a defined syntax convention. The set of constraints applying to a particular document is defined in a DTD.

Zusammenfassung

Web-Services – ein neues Paradigma in der Softwarearchitektur

Verschiedene Initiativen im Bereich verteilter Web-Applikationen haben in den letzten Monaten die Aufmerksamkeit der Entwicklergemeinde und IT-Verantwortlichen auf sich gezogen. So hat Microsoft mit der .NET-Initiative eine Revolution auf dem Gebiet der Internet-Dienste angekündigt, basierend auf offenen Standardformaten einerseits und proprietären Lösungen andererseits. Ähnliche Programme sind auch von SUN (ONE), HP (E-Speak) und anderen vorgestellt worden. Hinter all diesen Initiativen steckt das alte Bedürfnis, Applikationen dynamischer und flexibler aus bestehenden, verteilten Komponenten bauen zu können, um dadurch schneller und kostengünstiger Dienstleistungen und Produkte auf den Markt bringen zu können. Aufgrund der Tatsache, dass dieser neue Trend sich auf De-facto-Standards wie XML für den Datenaustausch und SOAP für die Verknüpfung von Funktionen stützt, ergibt sich ein grosses Potenzial, sowohl innerhalb von Firmen (Verbindung von Kernsystemen) wie auch zwischen Firmen (Marktplätze) mit dem neuen Paradigma Web Services Produkte flexibler und schneller auf den Markt zu bringen.