Analysis and Selection of Web Service Technologies

Viktorija Boltunova[1], Natalya Prokofyeva[2]
RTU student[1], RTU asoc. prof, dr.sc.ing.[2], Riga Technical University, Faculty of Computer Science and Information Technology. Address: Kalku iela 1, Riga, LV-1658, Latvia.

Abstract. The main objective of this article is to provide insights on how applications can interact with each other on the Web using Web services. The article is devoted to Web services and their basics, as well as to technologies such as REST-compliant (RESTful) and arbitrary (XML, SOAP, WSDL, UDDI, XML-RPC) that are used to implement them.

Keywords: SOAP, Web service, REST.

I. INTRODUCTION

The Internet has become a significant factor in business and public life. With the use of Internet technologies many problems can be solved. However, the Internet combines many different platforms and the information is saved in the various data sources. Linking of heterogeneous data communications as well as providing a method which allows obtaining it in a convenient form for further processing is an actual problem. The concept of Web services is designed to solve this problem of heterogeneous systems based on open standards association and integration [1], [2].

Web service is a service that is available on the Internet through a special program. For example, the most common services are search engines, web hosting (services that enables people and organizations to make their websites accessible via the World Wide Web), e-mail, an information storage (files, bookmarks), calendar, etc. An important feature of the Internet service is that the service is not dependent on the Internet service provider, browser or computer – it can work with data anywhere in the world where there is Internet access. Web services are based on open standards and protocols (SOAP, XML-RPC, REST, etc.). Web service is a way of how applications can interact with each other in the Web. Applications can use different platforms and be written in different programming languages. Thanks to the fact that Web services are developed to ensure machine collaboration in the Web (Machine-to-Machine approach) each service interface can be described using a machine-readable format

.

II. MAIN IDEA

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. We can identify two major classes of Web services:

- REST-compliant Web services, in which the primary purpose of the service is to manipulate XML representations of Web resources using a uniform set of "stateless" operations; and
- Arbitrary Web services, in which the service may expose an arbitrary set of operations [3].

Fig. 1 shows the overall Web service architecture. There can be three instances identified that interact within a Web service:

- service requestor;
- service provider;
- service broker.

The service provider sends the WSDL file to the UDDI (Universal Description Discovery & Integration). UDDI is an open source project that enables organizations to publish a Web service description (WSDL) so that it could be found by other organizations and integrated into their system. Service requester contacts the UDDI to find out what data provider he needs and then contacts the service provider via SOAP protocol. The service provider validates the service request and sends structured data in an XML file, using the SOAP protocol. This XML file would be validated again by the service requester using an XSD file.

ISSN 1691-5402

http://dx.doi.org/10.17770/ etr2017vol2.2540

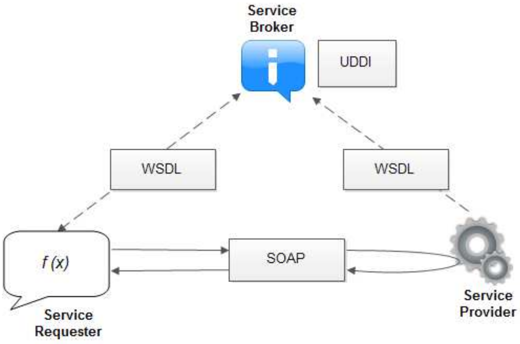

Fig. 1 Web service architecture.

III. WEB SERVICE TECHNOLOGIES

Web service architecture includes a lot of layers and related technologies. There are many ways how to visualize these technologies, as there are many ways how to build and use Web services. **Error! Reference source not found.** represents one of the technology families. XML, WSDL, SOAP, XML-RPC and UDDI technologies will be considered more detailed.

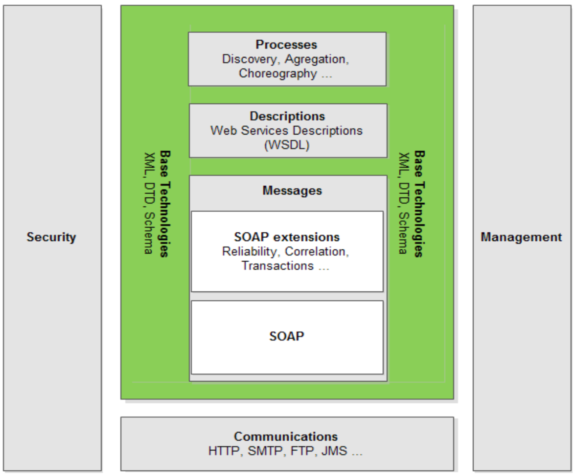

Fig. 2 Web service architecture stack.

A. XML

XML stands for *eXtensible Markup Language* which was recommended by the World Wide Web Consortium (W3C) on February 10, 1998. The XML specification describes XML documents. XML was designed as a formal language with a simple syntax that programs and humans can use to create and handle documents that will mostly be used on the Internet. The language is called extensible because it does not define the markup used in the documents: the developer is free to create the markup in accordance with the needs of a problem domain, being limited only by the syntax rules. A combination of simple formal syntax, convenience to the users and scalability, as well as being based on the Unicode encoding which is applied to represent the content of the documents, led to widespread use of XML, as well as using many other derivative specialized XML-based languages in different software tools. XML-document is a plain text file in which data elements are created with the help of special markers, which define the sequence and nesting of the document structure and its content. The main advantage of XML documents is that with a relatively simple way of creating and processing (plain text can be edited by any test processor and processed by standard XML parser) they allow to create structured information, which is well understandable to computers.

XML does not do anything by itself. The XML format was created to structure, store and transport information. Below is a note from John to Eve, which was saved as XML.

```
<note>
  <to>Eve</to>
  <from>John</from>
  <heading>Question</heading>
  <body>Do you know what is XML?</body>
</note>
```

The note contains information about sender and recipient. It also has a title and a message body. But still this XML document does not do anything. XML is only information that is "wrapped" in tags. Developer should write software to send, receive, store and display the message which is shown in Fig. 3 [4].

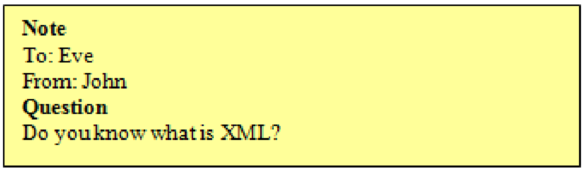
Note
To: Eve
From: John
Question
Do you know what is XML?

Fig. 3 Example of a note.

B. WSDL

WSDL stands for *Web Services Description Language* which was recommended by the World Wide Web Consortium (W3C) on June 26, 2007. This standard was developed by IBM, Microsoft and webMethods. Each of these three companies had their own approach in developing of a standard for describing Web-services: IBM created NASSL, Microsoft developed SCL and webMethods company came up with WIDL. WSDL is based on XML and describes Web services as endpoint nodes or port groups. Abstract port and message definitions are separated from specific implementations. This allows the use of these definitions again elsewhere. The port is defined by associating a binding to a specific Web address. WSDL is often used in combination with SOAP and XML Schema to provide Web service availability on the Internet. When connecting to a Web service, a client program can read the WSDL file and determine which operations are available. Each WSDL document contains several items which belong to one of the following categories as shown in Fig. 4 (category names as in WSDL 1.1 and

WSDL 2.0 specifications are given in the parentheses) [5], [6]:

- Service (Service / Service) - service can be considered a system function that is available with the help of web protocols.
- Port (Port/Endpoint) - the port defines a web service address. Typically, it is a string containing a HTTP URL address.
- Binding (Binding / Binding) - defines the interface and SOAP binding type (Remote Procedure Call/Document). Binding section also defines operations.
- Port Type (portType/Interface) - describes the Web service operations that can be executed and messages for operation execution.

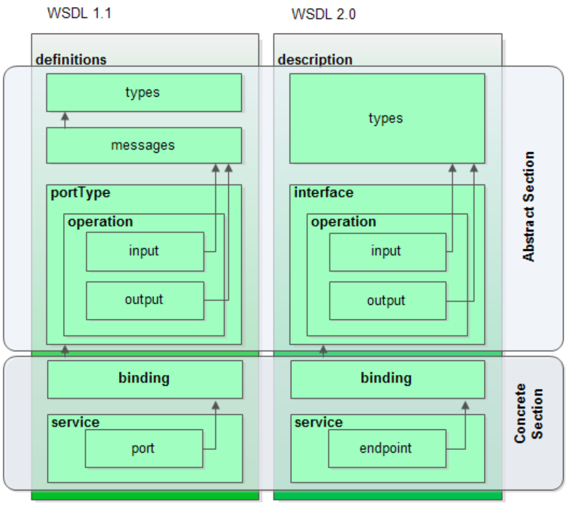

Fig. 4 WSDL 1.1 and WSDL 2.0 schematic diagram of concepts.

The main structure of a WSDL document looks like this:

```
<definitions>
<types>
  data type definitions
</types>
<message>
  definition of the data being communicated
</message>
<portType>
  set of operations
</portType>
<binding>
  protocol and data format specification
</binding>
</definitions>
```

C. SOAP

SOAP stands for *Simple Object Access Protocol* which was recommended by the World Wide Web Consortium (W3C) on June 24, 2003 (version 1.2). SOAP is an application communication protocol which is based on XML. SOAP is a format for sending and receiving messages. The main advantage of SOAP is that it is platform independent so it provides a way to communicate between applications running on different operating systems, with different technologies and programming languages. SOAP can be used with any application layer protocol: SMTP, FTP, HTTP, HTTPS, and others.

SOAP specification defines a messaging framework that includes the following components [7], [8]:

- SOAP processing model - describes the rules that define processing of a SOAP message.
- SOAP extensibility model - defines SOAP features and modules.
- SOAP underlying protocol binding - a framework which describes exchange of SOAP messages between SOAP nodes.
- SOAP Message Construct - describes SOAP message structure.

SOAP messages are using XML format which makes them easy to transfer, read and process. Fig. 5 shows that SOAP message consists of an envelope which is basically a container. The envelope contains a header and a body. The header contains information about how the recipient should handle received SOAP message. SOAP message body part contains information which is necessary for instruction execution.

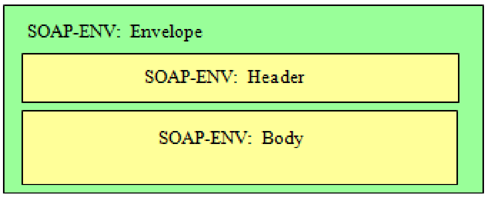

Fig. 5 SOAP message structure.

SOAP message skeleton looks like this:

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml;
charset=utf-8
Content-Length: 299

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-
envelope">
  <soap:Header>
  </soap:Header>
  <soap:Body>
  </soap:Body>
</soap:Envelope>
```

D. XML-RPC

XML-RPC stands for *Extensible Markup Language Remote Procedure Call* - remote procedure call standard/protocol which uses XML to encode its messages and HTTP as a transport mechanism. Fig. 6 shows that XML-RPC is a bridge between service requestor and service provider. In spite of the fact that XML-RPC is the ancestor of SOAP, it differs with its outstanding ease of use. XML-RPC as well as any other interfaces of Remote Procedure Call (RPC) defines a set of standard data types and commands that the programmer can use to access the functionality of a different program located on another computer in the network. XML-RPC is not intended to solve global tasks as SOAP but it is widely used in Web development.

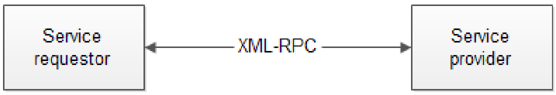

Fig. 6 XML-RPC conception.

Table I represents some of the characteristics that define differences between XML-RPC and SOAP. Data shows that data types in both protocols are the same. But in XML-RPC it is not possible to specify names for arrays, structures (all structures and arrays are "anonymous") and developer-defined encoding.

In light of current pace of development of technologies and standards, especially the Web, the point - the ease of development and practical application is becoming more important.

Table I
Comparison of SOAP and XML-RPC main characteristics

Characteristics	XML-RPC	SOAP
Scalar data types	+	+
Structures	+	+
Arrays	+	+
Named arrays and structures	-	+
Developer defined encoding	-	+
Developer-defined data types	-	+
Detailing of errors	+	+
The ease of development and practical application	+	-

Most programmers and specifications developers agree that:

- if there is a need for a system that is dealing with complex logic, transferring large complex data structures, if there is a need for complete information about the client and request should contain instructions about its processing – it is preferable to use SOAP;
- if data are relatively simple and applications must run on multiple platforms and languages, if performance is important and system logic does not require complicated commands - it is preferable to use XML-RPC [9].

E. UDDI

UDDI stands for *Universal Description Discovery & Integration*. UDDI is a cross-platform software based on XML. It is an open source project sponsored by OASIS, which allows organizations to publish descriptions of Web services (WSDL) for them to be searched by other organizations and integrated into their systems. It also determines how services or applications communicate over the Internet. UDDI was originally proposed as a primary Web service standard. It is designed for surveying by SOAP messages and for providing access to Web Service Description Language (WSDL) documents that describes bindings of protocols and message formats required to interact with Web services listed in its directory.

In Fig. 7 is shown how a business publishes services to the UDDI registry. Then a client finds the service in the registry and receives service binding information. Finally, the client uses received information to invoke the service. A business or a company can register three types of information into a UDDI registry.

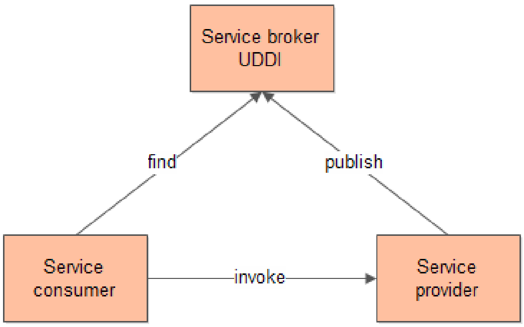

Fig. 7 UDDI registry.

Registration in the UDDI consists of following components:

- white pages - address, contact and known identifiers;
- yellow pages - industrial categorization based on standard taxonomies;
- green pages - technical information about the services available in the business.

White pages provide information about the service provider, for example a company name or description of services (probably in several languages). Using this information, it is possible to find a service that some information is already known about.

Yellow pages contain the classification of a business based on standard taxonomies. Since a business can provide a number of services there may be several yellow pages (each describing a service) related to a single white page (giving general information about the business).

Green pages are used to describe the method of gaining access to Web services and information about linked services. Part of the information is associated with Web services - such as address of the service and its parameters as well as links to interface specifications. Other information is not directly related to the Web service - it includes e-mail, FTP, CORBA, and service phone numbers. Since a Web service can have multiple bindings (as defined in its WSDL description) it may have multiple green pages because each binding will need access to various pages [10].

F. RESTful

REST stands for *Representational State Transfer* - architectural style of interaction between distributed application components in the network. REST is an alternative to RPC. The term “RESTful” is used for Web services that are based on REST architecture. REST is not a standard by itself, majority of RESTful implementations use standards such as HTTP, URL, JSON and XML. Although World Wide Web is based on this concept - the term “REST” was introduced only in 2000 by Roy Fielding who is one of the HTTP protocol founders.

REST offers HTTP-methods for developers to use. This basic REST design principle establishes an unambiguous equivalence between such operations as create, read, update and delete (CRUD) and HTTP-methods. According to this equivalence:

- to create a new resource on the server is used POST;
- to access a resource is used GET;
- to update a existing resource or create a new resource on the server is used PUT;
- to delete a resource is used DELETE [11].

Fig. 8 shows that REST is based on client-server interactions. The client application makes a request that is translated to a RESTful HTTP request. This request is started like any other HTTP transaction – from a client to a server. The server processes the request and responds accordingly.

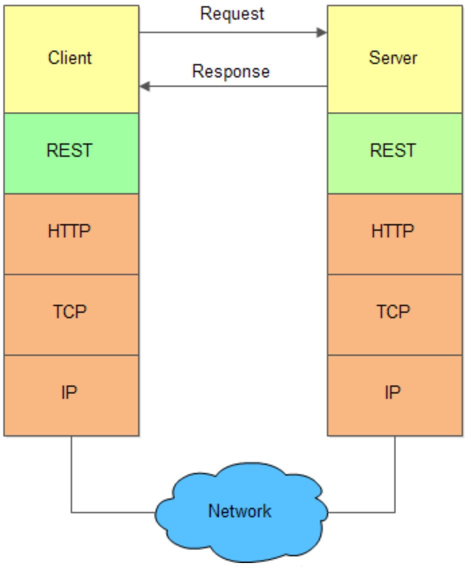

Fig. 8 Architecture of RESTful interactions.

IV. RELATED WORK

In our earlier work, we proposed various PHP programming framework [12] (CakePHP2, CodeIgniter, Symfony2, Yii, Phalcon) popularity overview and comparison using various criteria. Based on this study obtained data two frameworks were selected for deeper analysis - Symfony2 and PhalconPHP. There was offered a description of the architecture and main features of selected frameworks (routing, template engine, etc.). During framework comparison a performance test was developed with a goal to determine performance and effectiveness of frameworks for the same task. For performance testing a „Ticket Reserving System” cashier list section was selected. Tests were performed using ab.exe (Apache Benchmark) tool that comes along with the Apache Web server. Based on the comparison results, recommendations were made which allows Web developers to select a framework for the creation of a real Web project.

Although both Web service technology classes can be used for Web systems development, paper authors decided to develop Web site improved version using the REST architectural style. REST architecture choice is based on the fact that the protocol separates the user interface from the server and data storage, which improves interface portability to other types of platforms and increases the scalability of the project. Also the key aspect of REST architecture is that it's stateless (without status saving). This feature makes RESTful architecture secure and helps to improve the scalability.

Firstly, in order to improve the developed website capabilities, it was decided to use a Web service that will provide access to information about currency exchange rates. Currently the feature to book tickets is implemented on the website and registered users can see the prices for each performance. However, if a user from different country registers on the website, he will be more comfortable to look at the prices in his national currency. The Web service currencylayer.com was used to provide the information about currency exchange rates to the website. This Web service allows to use real-time REST API and retrieve the exchange rate history of the worlds 168 currencies and precious metals in the JSON format that is compatible with any application. Secondly, in order to improve the developed website capabilities, it was decided to use a Web service that will provide access to the information about the weather. Such opportunity will allow theatre visitors to prepare for unexpected weather changes. In order to implement such a feature, the Web service owm.io/weather-api was used, which aims to collect data on weather conditions around the world and make them more open and accessible to everyone.

V. CONCLUSIONS AND FUTURE WORK

The paper provides information about Web service basics and main technologies. In the paper REST-compliant (RESTful) and arbitrary (XML, SOAP, WSDL, UDDI, XML-RPC) Web service technologies that can be used in development of a Web system are described, also the description is provided about two Web services - currencylayer.com and owm.io/weather-api that were used to supplement developed website capabilities, and provide access to information on exchange rates and weather. The selected Web services are based on a REST architectural style, which nowadays is used more and more often in web-based technologies.

To continue the research, it is necessary to improve the Web system [12] which was developed using Symfony2 and PhalconPHP frameworks further by implementing Web services described in this paper to apply theoretical knowledge in practice.

REFERENCES

[1] C. Pautasso, O. Zimmermann and Leymann F., *RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision,* Beijing, 2008.

[2] E. Cavanaugh, *Web services: Benefits, challenges, and a unique, visual development solution,* Altova, 2006.

[3] Web Services Architecture. [Online]. Available: https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest. [Accessed: October 3, 2016].

[4] Introduction to XML. [Online]. Available: http://www.w3schools.com/xml/xml_whatis.asp. [Accessed: October 3, 2016].

[5] Web Services Description Language (WSDL) 1.1. [Online]. Available: http://www.w3.org/TR/wsdl. [Accessed: October 3, 2016].

[6] Web Services Description Language (WSDL) Version 2.0 Part 0: Primer. [Online]. Available: http://www.w3.org/TR/wsdl20-primer. [Accessed: October 3, 2016].

[7] Simple Object Access Protocol (SOAP) 1.1. [Online]. Available: https://www.w3.org/TR/2000/NOTE-SOAP-20000508. [Accessed: October 3, 2016].

[8] XML SOAP. [Online]. Available: http://www.w3schools.com/xml/xml_soap.asp. [Accessed: October 3, 2016].

[9] Лозовюк А., *"XML-RPC: вызов процедур посредством XML"*. Издательский Дом "КОМИЗДАТ". [Online]. Available: http://citforum.ru/internet/xml/xml_rpc. [Accessed: October 3, 2016].

[10] UDDI – Overview. [Online]. Available: https://www.tutorialspoint.com/uddi/uddi_overview.htm. [Accessed: October 8, 2016].

[11] RESTful Web Services – Introduction. [Online]. Available: https://www.tutorialspoint.com/restful/restful_quick_guide.htm. [Accessed: October 3, 2016].

[12] Prokofjeva N., Boltunova V., *Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems.* Procedia Computer Science, 2017, Vol.104, pp.51-56. ISSN 1877-0509. Available: https://doi.org/10.1016/j.procs.2017.01.059