EAI Endorsed Transactions on Scalable Information Systems Research Article ICST.ORG

QoS-Based Architecture for Discovery and Selection of suitable Web Services Using Non-functional properties

Sarathkumar Rangarajan[1,*] and Ravindra Krishna chandar [2]

[1] Centre for Applied Informatics, College of Engineering and Science, Victoria University, Melbourne
[2] Paavai Engineering College, India

Abstract

Web Services are the most emerging distributed applications published in the public registries. Web service are can be discovered by both functional properties and non functional properties. Due to the rapid Web development, there are number of functionally similar Web Services published by different vendors. The functional property based web service discovery is cannot be done with accuracy. So client can find the best Web Services by taking the non-functional criteria such as Quality of Service (QoS). However, most of clients are not experienced enough to acquire the best selection of Web Service based on its described QoS properties. In this paper we are proposing a client request message structure and broker architecture to find the best Web Service. First the broker will get Web Service client's requirement message along with QoS criteria, and then it will retrieve the functionally similar web service. The broker will use an efficient mechanism to rank the Web Services based on the client's message as well as the QoS properties which being confirmed by the broker architecture, If any tie situation happens in the ranking of the web service we will use the previous usage history of the web service to select the best web service which is matching with the client's request message.

Keywords: Web service, SOA, Discovery & selection, UDDI, QoS

Received on 05 September 2016, accepted on 06 September 2016, published on 19 January 2017

doi: 10.4108/eai.18-1-2017.152102

1. Introduction

Web Services (WSs) are segmental, self-defining, and slackly coupled software applications that can be publicized, found, and used across the Internet using a set of standards such as SOAP, WSDL, and UDDI [1]. It can be accessed over the network through the standardized XML messages. Though we are in the era of Internet of Things, there are number of firms developed Web Services and made available to the users. Web Services can be found either by functional criteria or non-functional criteria. Since there are many functionally alike Web Services available, the functional criteria is not an optimal one for the client to find the best Web Service. The ultimate solution for this problem is to use the non-functional criteria such as Quality of Service (QoS). By QoS, we refer to non-functional properties of Web Services some of them are,

Availability: Availability is the quality aspect of whether the Web Service is present or ready for immediate use.

Accessibility: Accessibility is the quality aspect of a Service that represents the degree it is capable of serving a Web Service request.

Integrity: Integrity is the quality aspect of how the Web Service maintains the correctness of the interaction in respect to the source.

Performance: Performance is the quality aspect of Web Service, which is measured in terms of throughput and latency. Higher throughput and lower latency values represent good performance of a Web Service.

Regulatory: Regulatory is the quality aspect of the WS in conformance with the rules, the law, compliance with standards, and the established Service level agreement.

[*]Corresponding author. Email: sarathkumar.rangarajan@live.vu.edu.au

Security: Security is the quality aspect of the Web Service of providing confidentiality and non-repudiation by authenticating the parties involved, encrypting messages, and providing access control.

However, most of clients are not skilled sufficiently to get the best Web Service based on its Web Service Description Language (WSDL). They simply trust the information published by the provider; however most of Web Service providers do not guarantee and assure the level of QoS offered by their Web Services. To address the above said issues, this paper proposes a Web Service discovery & selection architecture that contains an extended UDDI to accommodate the QoS information, and WS-QoS Broker to assist the Web Service discovery.

The broker based selection architecture uses the concept of middleware (broker) for QoS aware Web Service publishing and selection mechanisms. In broker based architecture, the broker is a critical architectural component of interaction for the requester and provider towards dynamic Web Service selection and publishing. The functionality of the broker is to select the most suitable Web Service for the Requester that satisfies his QoS constraints and preferences. The other functionalities of the broker may include QoS publishing, QoS verification & certification and QoS management & monitoring.

The rest of the paper is organized as follows, Section 2 contains the existing research on Web Service QoS broker. Section 3 describes our contributions on how the Web Service client can send their requirement to the broker QoS broker architecture and selection mechanism of the best Web Service. In section 4 describes the related work. Section 5 presents the results and discussion and finally Section 6 carries conclusions.

2 Web Service Search Architecture

The Existing Web Services search architecture is based upon the interactions between three roles: service provider, service registry and service requestor. The interactions contains publish, find and bind operations. In a typical scenario, a service provider hosts a network-accessible software module (an implementation of a Web service).

The service provider defines a service description for the Web service and publishes it to a service requestor or service registry. The service requestor uses a find operation to retrieve the service description locally or from the service registry and uses the service description to bind with the service provider and invoke or interact with the Web service implementation. Service provider and service requestor roles are logical constructs and a service can exhibit characteristics of both.

For an application to retrieve of Web Services, three behaviours must take place that are, publication of service descriptions, lookup or finding of service descriptions, and binding or invoking of services based on the service description. These behaviours can occur singly or iteratively. In detail, these operations are:

Publish: To be accessible, a service description needs to be published so that the service requestor can find it. Where it is published can vary depending upon the requirements of the application (see “Service Publication” for more details).

Find: In the find operation, the service requestor retrieves a service description directly or queries the service registry for the type of service required (see “Service Discovery” for more details).

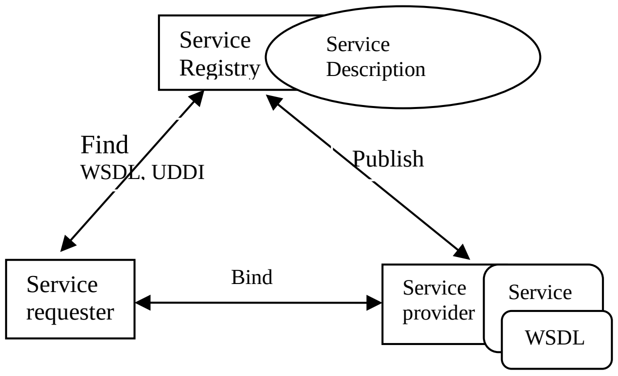

Figure 1. Web Service Search Architecture

Bind: Eventually, a service needs to be invoked. In the bind operation the service requestor invokes or initiates an interaction with the service at runtime using the binding details in the service description to locate, contact and invoke the service.

3 Our Contribution

In this Research we are proposing a client request message structure and broker architecture to find the best Web Service.

Figure 2 represents the broker architecture for the proposed approach. Broker itself a webservice. This enables the architecture deployment in restricted and open environments. The architecture consists of the basic web service model components like the web service provider, web service consumer and the UDDI registry. In addition, UDDI registry has the capability to store QoS information using tModel data structure and a WS-QoS Broker component.

3.1 Web Service Request Message Structure

Web Service clients need to give the functional and non-functional criteria for the target Web Service in a standardized XML message consists of functional requirement and non-functional requirements. For each quality measures the client has to give the weight value, because the client may specify many numbers of quality aspects but the broker must understand the key attribute for the client, so that the QoS Broker can select the best suitable Web Service for the Web Service client.

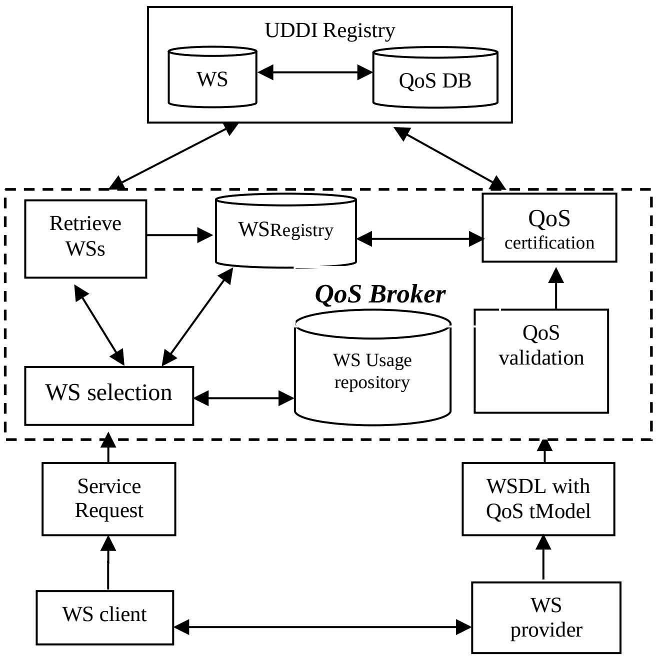

Figure 2. QoS based web service selection Broker

3.2 Data structure for the QoS information

The QoS properties of a Web Service can be represented using the data structure called tmodel [8].

```
<tModel >
 <name> QoS Information for Stock Quote Service
 </name>
<overviewDoc>
   <overviewURL>
  http://<URL describing schema of QoS attributes
   </overviewURL>
</overviewDoc>
<categoryBag>
  <keyedReference
  tModelKey="uddi:uddi.org:QoS:Availability"
  keyName="Availability"
  keyValue="99.9%" />
  <keyedReference
  tModelKey="uddi:uddi.org:QoS:Throughput"
  keyName="Average Throughput"
  keyValue=">10Mbps" />
</categoryBag>
</tModel>
```

Figure 3. tModel data of a Sample webservice

The role of a tModel is to register categorizations, which provides anextensible mechanism for adding property information toa UDDI registry. It can be used to provide QoS information on binding Templates.

Figure 3 shows an example for tModel of a webservice. In the tModel forquality of service information for the binding templatethat represents a Web Service deployment is generated torepresent quality of service information. Each QoS metric, such as average response time or average throughput is represented by a keyedReference that is a general-purpose structure for a name-value pair, on the generated tModel.

3.3 Service Publisher

The service publisher component communicates with the service provider and the UDDI registry. The web service provider registers the business and web service related information with the service publisher. It also gets the specific QoS property values of web services from providers. Once the QoS property values and other information are obtained from the provider it is hand over to the Verifier and Certifier component. The QoS information is verified and certified before publishing it in the UDDI registry.

3.4 QoS Verifier and Certifier

QoS Verification & certification is the key component of the WS-QoS Broker that performs the verification of the QoS information supplied by the service provider and issues a certificate to the service provider through the service publisher. This QoS certificate assures that the QoS offered by the provider conform to their descriptions. The service provider initiates the verification process through the service publisher by supplying the QoS property values. The verifier is provided with the WSDL document and additional information about resources available at the provider’s platform.

The verifier performs the testing of the service URI, the XML schema definition, the service binding information and the availability of all operations described in the service interface. Verifier also performs the verification of the QoS information introduced in the service interface. The QoS verification is conducted through a set of test cases Generated by the verifier. For each test, additional information like server capacity, network bandwidth about the provider and its web services are needed. The four QoS parameters (Response Time, Availability, Throughput, and Price) are also verified. The verification process is done in three levels: General web services information verification, WSDL content verification and QoS verification.

A web service is said to be compliant with a given level when it passes the corresponding tests described in the verification document. Based on this, web services can be classified into three classes. Class A includes web services for which all verification tests have succeeded.

Class B includes web services for which more than 80% of the verification tests have succeeded. Class C contains the services for which most of the verification scenarios have failed. Once the verification process is completed successfully, the certification process is initiated.

The certifier issues a certificate to the service provider through the service publisher which indicates that the offered QoS conform to their descriptions. The main responsibility of the certifier is to certify the web services and their provided QoS. A copy of the certificate sent to the service provider, which is also stored in the WSS for future use. The certificate includes information such as certificate number, certificate issue date, number of years in business and service location. In case, if the certificate cannot be issued, feedback will be sent to the provider. After the QoS certification process, the service publisher can register the functional description of the web service and the certified QoS information with the UDDI registry.

3.5 Discovering the Web Service

WSDLs for each service are published in a UDDI registry as a tModel. The find_tModel message will return a list of tModel keys. A specific service interface description is retrieved using the get_tModelDetail message. After a tModel has been retrieved, the overviewURL can be used to retrieve the contents of the WSDL service interface document. Additional keyedReferences can be added to the categoryBag to limit the set of tmodel that are returned in the response to this find request.

```
<find_tModel generic="1.0" xmlns="urn:uddi-org:api">
  <categoryBag>
    <keyedReference
    tModelKey="UUID:DB77450D-9FA8-D14E384"
    keyName="Stock market trading services"
    keyValue="84121801"/>
  </categoryBag>
</find_tModel>
```

Figure 4. Sample find_tModel

3.6 Flow of Qos Broker

The design of the broker as follows, Client will search for the web service with search key along with required QOS parameters. Broker will search the web service registry for client's query. Then Broker will place the retrieved web service in the retrieved web service repository. After that the broker using the min max normalization tree to find the best web service. Once the Broker found the best web service, it adds the entry of the web service which is being served to the client.

Finally Negotiation and binding of the web service will be done. If any two or more web service getting tie between them when assigning the QoS rank, we can use the usage counter value to find the most used web service and that will served as the best web service for the client's request.

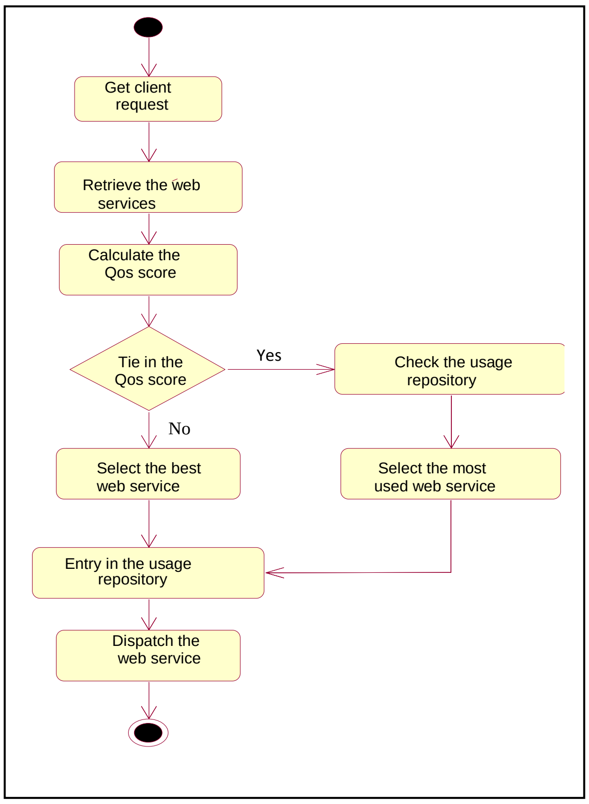
Get client request
Retrieve the web services
Calculate the Qos score
Tie in the Qos score
Yes
Check the usage repository
No
Select the best web service
Select the most used web service
Entry in the usage repository
Dispatch the web service

Figure 5. Flow chart for QoS broker

4 Related work

The Web Service broker will get the quality criteria for the Web Service from the requester's message. Then the broker will discover the Web Services from the UDDI registry by using the tmodel_find message. Then it will place all the Web Services in the Web Service storage.

Tree Construction & Ranking

The Web Service broker obtain both functional and non-functional criterion for the target Web Service from the requester's message .Then the broker will discover the Web Services from the UDDI registry by using the tmodel_find message and placed them in the local Web Service repository.

The Next step in the selection process is to sort the web services based on the client's request. For that we used the min-max normalization technique and weighted AND-OR tree to obtain the Quality Constrain Tree (QCT) which is explained in [1]. In the Weighted AND-OR tree every edge between parent and child node is labelled with non-negative real number in an interval (0, 1) such that for any parent node the sum of edge labels (weights) of all child nodes is equal to one i.e. for any parent node *P* with *C (2_C_N)* child nodes, the sum of edge weights *WPCi (1_i_C)* is equal to 1. Each leaf node having the quality criteria of the client and all the obtained Web Services in the local registry matched with the leaf node. The web services which all are passing through the leaf node criteria will be given a score using the formula (1) & (2).

wss_n	- nth web service's QoS Score.
ws_{first}	- first web service in the descending order in the leaf node.
ws_{last}	- Last web service in the descending order in the leaf node.

In "<" conditioned leaf node :

$$wss_n = \begin{Bmatrix} \frac{2(ws_{last} - ws_x)}{10} \\ wss_{first}=1 \\ wss_{last}=0 \end{Bmatrix} x = 2,3,..last - 1 . \qquad (1)$$

In ">"conditioned leaf node:

$$wss_n = \begin{Bmatrix} \frac{2(ws_{first} - ws_x)}{10} \\ wss_{last}=1 \\ wss_{first}=0 \end{Bmatrix} x = 2,3,..last - 1. \qquad (2)$$

If the root node for the leaf nodes is a AND node then web services which all present in all the leaf nodes will be selected and new score will be calculated using weight of the particular leaf node. After constructing the tree the root node will be having the descending ordered list of the Web Services based on their QoS Score. In the top ranked one is the best Web Service based on the particular client's quality requirement. Once finding the best target Web Service we need to increment the usage count value of the particular web service in the Web Service repository. Then negotiation and binding process initiated by the interaction layer.

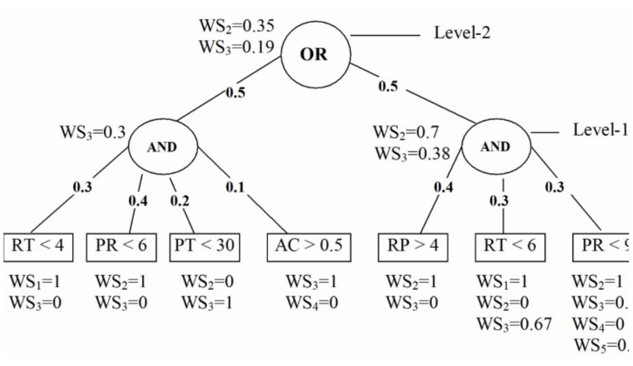

Figure 6. Example of the QCT

Figure 6 shows an example for a QCT with web service scores. In the QCT, if the root node having two Web Services having same QoS Score then broker need to check for another aspect to choose the best one. For this [2] defining four Web Service Provider qualities to compare and find the best Web Service in case of the Tie. In this research we use another method to find the best web service in the tie situation which is explained in the Figure 5.

We increment the count for the Web Service in the usage repository whenever a Web Service is happened to bind with a business client. So we can use this count value as the criteria for selection. Because the service which having the maximum count value selected as a best Web Service for many clients. In case of the tie in the QCT, the broker will obtain the count value of the both Web Service and select the maximum count valued Web Service as the best one.

5 Results and Discussion

The broker environment implemented using NetBeans IDE 7.1 through java. Where we have a Client interface is the starting point of the experiment. In this form client has to type their functional and non-functional required towards the required web service. Then we used XML packages to create the SOAP message of the client requirement then that will be sent to the broker application.

For the UDDI registry we used QWS-WSDLs Dataset Version 1.0 [17], [18], [19] for the web service list as well as the QoS properties of the particular web services. QWS Dataset Version 2.0 includes a set of 2,507 Web services and their QWS measurements. We used Mysql database to store the dataset and using XAMPP we made the Mysql available for the runtime environment.

Once the broker got the user, it will search the Mysql database for the functionally similar web service then it will make normalization as per the mathematical model. Then finally it returns the ranked web service list along with the QoS Score. Then the client gets the top ranked web service by clicking the button.

5.1 Performance Evaluation

While comparing the existing web service search based on functional keyword with the QoS based broker architecture the quality based search will give the optimal result with respect to the client's requirement.

Figure 7 shows the graph comparing the user rating against the number of functionally similar web services. In the functional requirement based search results will give the good result while the number of web service is low, it will give less user rating when the number goes on. But the QoS based search will give the optimal search result even though the number of web services increased.

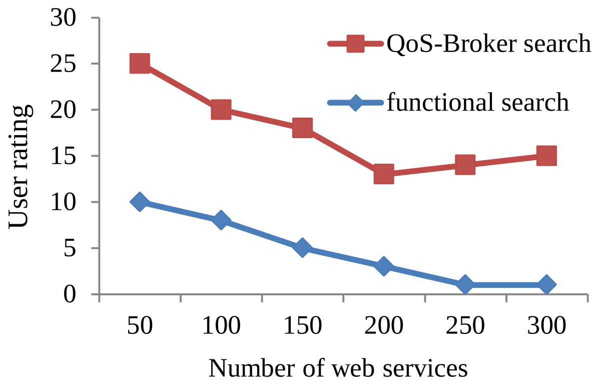

Figure 7. Comparison of user reviews with & Without QoS Broker based search

6 CONCLUSION

In this research we introduces a standard method to represent the Web Service client's requirement message and then we proposed QoS broker architecture to get the client's functional and non-functional requirements for webservice and broker discovered the Web Service from the local Web Services repository based on the functional requirement of the client, finally it will select the best suitable Web Service for the client's QoS requirements by using the min-max normalization technique. After the successful utilization of a webservice, the usage count variable will be incremented in the usage repository. Finally the broker will dispatch the best Web Service to the client. We compared QoS broker based WS search with functional based WS search methodology in a repository with 300 WSs. When the number of WSs increases our methodology provides better results and return best suitable webservice for the client. This research can be extended by composition for different types of webservices to create a Service Oriented Architecture for business needs.

References

[1] D'Mello, D.A., Ananthanarayana, V.S., Santhi, T " A QoS Broker Based Architecture for Dynamic Web Service Selection, Modeling & Simulation", 2008. AICMS 08. Second Asia International Conference.

[2] T.Rajendran, Dr.P.Balasubramanie, Resmi Cherian "An Efficient WS-QoS Broker Based Architecture for Web Services Selection" , 2010 International Journal of Computer Applications ,Volume 1 – No. 9.

[3] D'Mello, D.A., Ananthanarayana "Quality Driven Web Service Selection and Ranking", Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference.

[4] Reza Tabein, Mahdi Naser Moghadasi, Alireza Khoshkbarforoushha" Broker-based Web Service Selection using Learning Automata" , Service Systems and Service Management, 2008 International Conference .

[5] Tao Yu , Kwei-Jay Lin " A Broker-Based Framework for QoS-Aware Web Service Composition", e-Technology, e-Commerce and e-Service, 2005. EEE '05. Proceedings. The 2005 IEEE International Conference.

[6] Kyriakos Kritikos,Dimitris Plexousakis "Requirements for QoS-Based Web Service Description and Discovery", Ieee Transactions On Services Computing, Vol. 2, No. 4, October-december 2009

[7] Hai Liu, Weimin Zhang, Kaijun Ren , Zhuxi Zhang," A Novel Selection Approach for Transactional Web Services Composition ", 2010 Ninth International Conference on Grid and Cloud Computing 2010 IEEE.

[8] M.A.Serhani, R.Dssouli, A.Hafid, H.Sahraoui," A QoS broker based architecture for efficient Web Services selection", Proceedings of the IEEE International Conference on Web Services (ICWS'05) .

[9] Gwyduk Yeom, Taewoong Yun, Dugki Min ," A QoS Model and Testing Mechanism for Quality-driven Web Services Selection" , Proceedings of the Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems and Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06)

[10] Demian Antony D'Mello, Ananthanarayana V. S." A Review of Quality of Service (QoS) Driven Dynamic Web Service Selection Techniques" , 5th International Conference on Industrial and Information Systems, ICIIS 2010

[11] T. Rajendran, Dr.P. Balasubramanie ,"An Optimal Broker-Based Architecture for Web Service Discovery with QoS Characteristics" , International Journal of Web Services Practices, Vol. 5, No.1 (2010)

[12] Andrea D'Ambrogio., "A Model-driven WSDL Extension for Describing the QoS ofWeb Services", IEEE International Conference on Web Services (ICWS'06), 2006.

[13] Keller, A. and Ludwing. H., 2002, "The WSLA framework: Specifying and Monitoring Service Level Agreements for Web Services", IBM Research Report.

[14] Serhani, M.A., Dssouli, R., Hafid, A. and Sahraoui, H., 2005, "A QoS broker based architecture for efficient Web services selection". In Proc. of the IEEE Int'l Conf. on Web Services, IEEE CS, pages 113–120.

[15] Ziqiang Xu, Patrick Martin, Wendy Powley and Farhana Zulkernine, 2007, "Reputation Enhanced QoS-based Web services Discovery", IEEE International Conference on Web Services (ICWS 2007).

[16] Reza Tabein, Ali Nourollah," Dynamic Broker-based Service Selection with QoS-driven Recurrent Counter Classes", IEEE, 2008.

[17] Al-Masri, E., and Mahmoud, Q. H., "Discovering the best web service", (poster) 16th International Conference on World Wide Web (WWW), 2007, pp. 1257-1258.

[18] Al-Masri, E., and Mahmoud, Q. H., "QoS-based Discovery and Ranking of Web Services", IEEE 16th International Conference on Computer Communications and Networks (ICCCN), 2007, pp. 529-534.

[19] Al-Masri, E., and Mahmoud, Q.H., "Investigating Web Services on the World Wide Web", 17th International Conference on World Wide Web (WWW), Beijing, April 2008, pp. 795-804.