



ISBN	978-81-929866-5-4
Website	icca.co.in
Received	14 – March– 2016
Article ID	ICCA013

VOL	05
eMail	icca@asdf.res.in
Accepted	02 - April – 2016
eAID	ICCA.2016.013

# Applying Microservices in Webservices, with An Implementation Idea

J Sylvia Grace<sup>1</sup>, R Sreeranjani<sup>2</sup>, A Rubika<sup>3</sup>

<sup>1,3</sup>Assistant Prof, CSE, <sup>2</sup>Meenakshi Coll of Engineering, Chennai

**Abstract:** *Web Services are self-describing services that will perform well defined tasks and can be accessed through the web. Service Oriented Architecture (SOA) is an architecture paradigm that focuses on building systems through the use of different Web Services, integrating them together to make up the whole system.*

*The Micro-service style is an approach to develop a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms. Preferring Micro-services over others, it makes the work on both client and server side smooth. This paper explains about the emerging Micro-services - which is the practice of applying SOA principles at a small level of granularity. This implementation idea explains how the Anna University result webpage can be divided using micro-services for easy access of results to students.*

**Keywords:** *Web-services, Micro-service, SOA, Monolithic*

## INTRODUCTION

The big idea behind micro-services is to architect large, complex and long-lived applications as a set of cohesive services that evolve over time. It is even been called lightweight or fine-grained SOA and is different than traditional SOA. More importantly, it solves many of the problems that many organizations currently suffer from. The goal of decomposition is not to have tiny services but to address the problems and limitations of the monolithic (large) architecture.

### 1. Web Services

A Web service, is a method of communication between two applications or electronic devices over the World Wide Web (WWW). Also called as application services, these are services including a combination of programming and data, which are made available from a Web server for Web users or other Web-connected programs. Providers of Web services are generally known as application service providers. Users can access some Web services through a peer-to-peer arrangement rather than by going to a central server. Some services can communicate with other services and this exchange of procedures and data is generally enabled by a class of software known as middleware. Web services are also increasingly enabled by the use of the Extensible Markup Language (XML). [1] Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone, where

- XML is used to tag the data
- SOAP and REST are used to transfer the data
- WSDL is used for describing the services available
- UDDI is used for listing what services are available. [2]

This paper is prepared exclusively for International Conference on Computer Applications 2016 [ICCA 2016] which is published by ASDF International, Registered in London, United Kingdom under the directions of the Editor-in-Chief Dr Gunasekaran Gunasamy and Editors Dr. Daniel James, Dr. Kokula Krishna Hari Kunasekaran and Dr. Saikishore Elangovan. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honoured. For all other uses, contact the owner/author(s). Copyright Holder can be reached at copy@asdf.international for distribution.

2016 © Reserved by Association of Scientists, Developers and Faculties [www.ASDF.international]

**Cite this article as:** J Sylvia Grace, R Sreeranjani, A Rubika. "Applying Microservices in Webservices, with An Implementation Idea". *International Conference on Computer Applications 2016*: 64-69. Print.

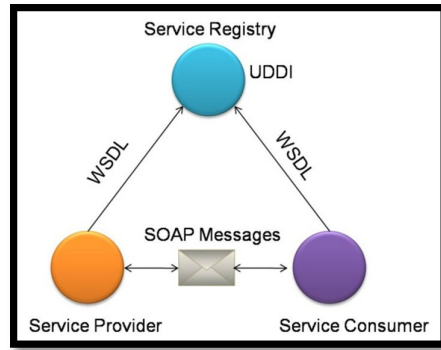


Figure 1: Web service

### 1.1 Soap and Rest

Web services are of two kinds:

1. Simple Object Access Protocol (SOAP)
2. Representational State Transfer (REST)

SOAP defines a standard communication protocol (set of rules) specification for XML-based message exchange. SOAP uses different transport protocols, such as HTTP and SMTP. [3]. SOAP defines its own security and is less preferred than REST. [4]

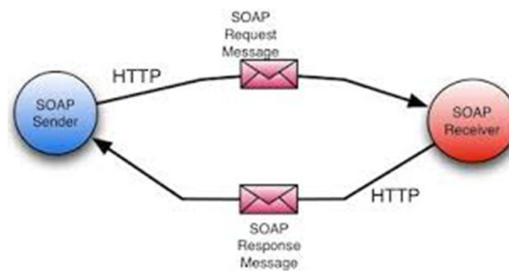


Figure 2: SOAP

REST describes a set of architectural principles by which data can be transmitted over a standardized interface (such as HTTP). A client can access the resource using the unique URI and a representation of the resource is returned. While accessing RESTful resources with HTTP protocol, the URL of the resource serves as the resource identifier and GET, PUT, DELETE, POST and HEAD are the standard HTTP operations to be performed on that resource. [4]

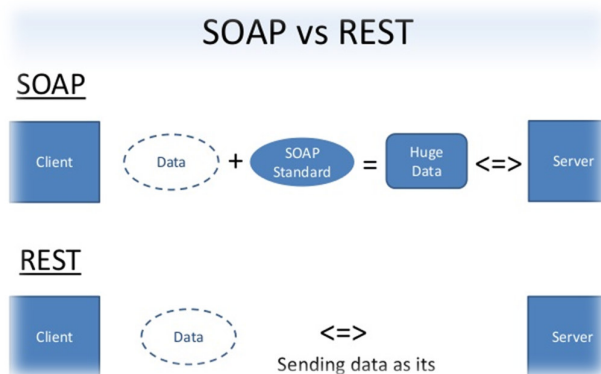


Figure 3: SOAP vs REST

**Cite this article as:** J Sylvia Grace, R Sreeranjani, A Rubika. "Applying Microservices in Webservices, with An Implementation Idea". *International Conference on Computer Applications 2016*: 64-69. Print.

## 2. SOA

Service oriented architecture (SOA) is the application of service requester/provider concepts to middleware technology. With its loosely coupled nature, it allows enterprises to plug in new services or upgrade existing services in a granular fashion. It brings better reusability of existing assets and lets you create applications that can be built on top of new and existing applications. [5]

SOA is essentially a collection of services which communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity.

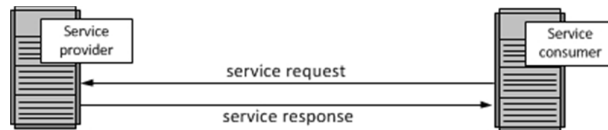


Figure 4: SOA Working

The benefit of implementing SOA with Web services is that you achieve a platform-neutral approach to access services and better interoperability as more vendors support more Web services specifications. [6]

## 3. Microservices

The emerging viewpoint on SOA is Microservices. Microservices is the practice of applying service-oriented architecture principles at a small level of granularity. These are self-contained and don't necessarily require an application server to host them. They enable you to implement much smaller deployable units, which then helps to push out updates or do features and capabilities much more quickly.

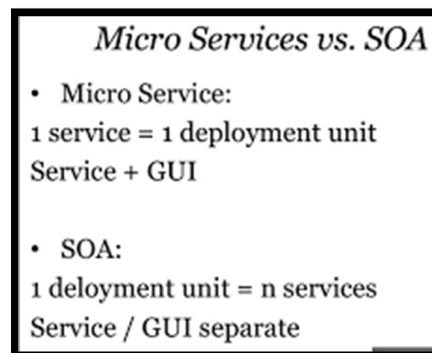


Figure 5: Microservices vs SOA

### 3.1 WHY Microservices in SOA?

Microservices are self-contained and don't necessarily require an application server to host them. These enables to implement much smaller deployable units, which then allows to push out updates or enhance features much more quickly.

Being just a small section of a much larger application, it helps to replicate and create multiple instances. Also, it spreads the load for just that one small piece of the application instead of having to do the entire system.[4]



Figure 6: Representing Microservices

**Cite this article as:** J Sylvia Grace, R Sreeranjani, A Rubika. "Applying Microservices in Webservices, with An Implementation Idea". *International Conference on Computer Applications 2016*: 64-69. Print.

#### 4. Monoliths

Monolithic servers are developed and deployed as a single unit. They increase the complexity and size and decrease the development, testing and deployment speed. This approach works well for relatively small applications. A large monolithic application can be difficult for developers to understand and maintain. To deploy changes to one application component you have to build and deploy the entire monolith, which can be complex, risky, time consuming, require the coordination of many developers and result in long test cycles.

This architecture also makes it difficult to trial and adopt new technologies. [5]

#### 5. Upper Hand of Microservices over Monoliths

The concept of breaking monolithic applications into smaller applications began with SOA, in terms of breaking applications into modules and then to smaller services, called Microservices. This approach is to break the apps into separate containers in a way that enables intra-application communication rather than inter-application communications. Thus, this concept coincides with continuous delivery practices, where the goal is to push out small units without having to create a monolithic system.[5]

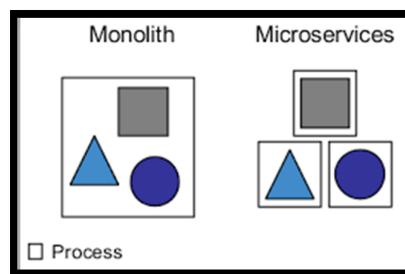


Figure 7: Differentiating Monolith & Microservices

#### 6. Enabling Microservices in Frequently used Websites

In the recent days, students encounter the problem while checking their results. Considering one such web page - The Anna University Result Web page - we implement the concept of microservices into it. Microservices are an approach to development of a single application composed of small services. The key to idea of microservices is independence. Each service is developed, tested and deployed separately from each other and every service runs as a separate process. The only relation between different microservices is data exchange accomplished through APIs they're exposing. [7]

##### 6.1 Execution of Suggested Example

The result page of Anna University intakes the student register number, date of birth and a security question in normal fashion. We have brought the idea of implementing microservices into it by adding 2 more input attributes to the result page, namely 'Place of college' and 'Department'. The 'Place of college' will be a drop down list which shows all the places where affiliated colleges are located.

Figure 8: Suggested new webpage

**Cite this article as:** J Sylvia Grace, R Sreeranjani, A Rubika. "Applying Microservices in Webservices, with An Implementation Idea". *International Conference on Computer Applications 2016*: 64-69. Print.

A student can give his/her register number, date of birth, place of college and a security question in the first page, and click Submit. This will validate all details and if they are correct, it directs to another page which shows all the college names located at the place given. The student can choose his/her college name, which directs to another page asking for the department. On giving this, the result of the student is easily located on the server, as many input queries filter the data's.

Since we direct the student to many pages unlike before, each page can be deployed using microservices.

That is, development of each page (referred by college name and place) can be done and stored in an individual fashion. This reduces load in both server side and client side. This methodology can even be applied to mobile applications, thus providing flexibility over monolith.

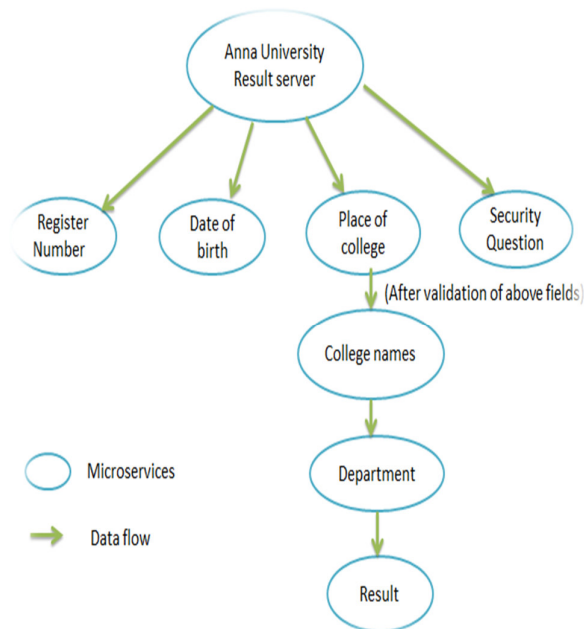


Figure 9: Information flow using Microservices

Now results can be viewed soon because the number of service requestors (students) get filtered continuously during progress of checking results. Thus, there will be around 552 microservices created for college names and around 15 microservices for Place of college. The number sounds large, but it produces great results. Its advantages are given below:

## 7. Positives

By deploying through microservice, both the service requestor and provider become advantageous. Few advantages are:

- Small and Simpler
- Independent Deployment:
- Independence in execution
- Improves fault isolation
- Reusability [8]

## 8. Implementing Microservices using Docker

Docker is an open-source project which aims to automate the deployment of applications inside portable containers that are independent of hardware, host operating system, and language. [9]

Containers allow a developer to package up an application with all of the parts it needs. The concept of Docker is all about making it easier to build, ship and run any application by using containers. [10]

- Build - Package your application in a container.

**Cite this article as:** J Sylvia Grace, R Sreeranjani, A Rubika. "Applying Microservices in Webservices, with An Implementation Idea". *International Conference on Computer Applications 2016*: 64-69. Print.

- Ship - Move that container from a machine to another.
- Run - Execute that container (the application) [11].



Figure 10: Docker

One of the major positive of Docker is, it suits best for microservices architecture. Containers also support micro services architecture. Each micro service can be deployed without interrupting the other micro services and containers provide an ideal environment for service deployment in meaning of speed, isolation management, and lifecycle. [12]

#### 9. REFERENCES:

1. <http://searchsoa.techtarget.com/definition/Web-services>
2. [http://www.webopedia.com/TERM/W/Web\\_Services.html](http://www.webopedia.com/TERM/W/Web_Services.html)
3. <http://searchsoa.techtarget.com/tip/REST-vs-SOAP-How-to-choose-the-best-Web-service>
4. <http://www.javatpoint.com/soap-vs-rest-web-services>
5. <http://www.infoq.com/articles/microservices-intro>
6. <http://www.javaworld.com/article/2071889/soa/what-is-service-oriented-architecture.html>
7. SOA is not webservices, <http://www.javaworld.com/article/2071889/soa/what-is-service-oriented-architecture.html>
8. <http://www.oracle.com/technetwork/articles/javase/soa-142870.html>
9. Breakup the monoliths, <http://searchcloudapplications.techtarget.com/feature/How-microservices-bring-agility-to-SOA>
10. <http://searchcloudapplications.techtarget.com/feature/How-microservices-bring-agility-to-SOA>
11. [https://en.wikipedia.org/wiki/Crash\\_%28computing%29](https://en.wikipedia.org/wiki/Crash_%28computing%29)
12. <http://technologyconversations.com/2015/01/07/monolithic-servers-vs-microservices/>
13. <http://www.infoq.com/articles/microservices-intro>
14. [https://www.ibm.com/developerworks/community/blogs/1ba56fe3-efad-432f-a1ab58ba3910b073/entry/microservices\\_architecture\\_containers\\_and\\_docker?lang=enIt's](https://www.ibm.com/developerworks/community/blogs/1ba56fe3-efad-432f-a1ab58ba3910b073/entry/microservices_architecture_containers_and_docker?lang=enIt's)