Web Service

* It is based on SOAP and return data in XML form.
* It support only HTTP protocol.
* It is not open source but can be consumed by any client that understands xml.
* It can be hosted only on IIS.

WCF

* It is also based on SOAP and return data in XML form.
* It is the evolution of the web service(ASMX) and support various protocols like TCP, HTTP, HTTPS, Named Pipes, MSMQ.
* The main issue with WCF is, its tedious and extensive configuration.
* It is not open source but can be consumed by any client that understands xml.
* It can be hosted with in the applicaion or on IIS or using window service.

WCF Rest

* To use WCF as WCF Rest service you have to enable webHttpBindings.
* It support HTTP GET and POST verbs by [WebGet] and [WebInvoke] attributes respectively.
* To enable other HTTP verbs you have to do some configuration in IIS to accept request of that particular verb on .svc files
* Passing data through parameters using a WebGet needs configuration. The UriTemplate must be specified.
* It support XML, JSON and ATOM data format.

Web API

* This is the new framework for building HTTP services with easy and simple way.
* Web API is open source an ideal platform for building REST-ful services over the .NET Framework.
* Unlike WCF Rest service, it use the full feature of HTTP (like URIs, request/response headers, caching, versioning, various content formats)
* It also supports the MVC features such as routing, controllers, action results, filter, model binders, IOC container or dependency injection, unit testing that makes it more simple and robust.
* It can be hosted with in the application or on IIS.
* It is light weight architecture and good for devices which have limited bandwidth like smart phones.
* Responses are formatted by Web API’s MediaTypeFormatter into JSON, XML or whatever format you want to add as a MediaTypeFormatter.

To whom choose between WCF or WEB API

* Choose WCF when you want to create a service that should support special scenarios such as one way messaging, message queues, duplex communication etc.
* Choose WCF when you want to create a service that can use fast transport channels when available, such as TCP, Named Pipes, or maybe even UDP (in WCF 4.5), and you also want to support HTTP when all other transport channels are unavailable.
* Choose Web API when you want to create a resource-oriented services over HTTP that can use the full features of HTTP (like URIs, request/response headers, caching, versioning, various content formats).
* Choose Web API when you want to expose your service to a broad range of clients including browsers, mobiles, iphone and tablets.