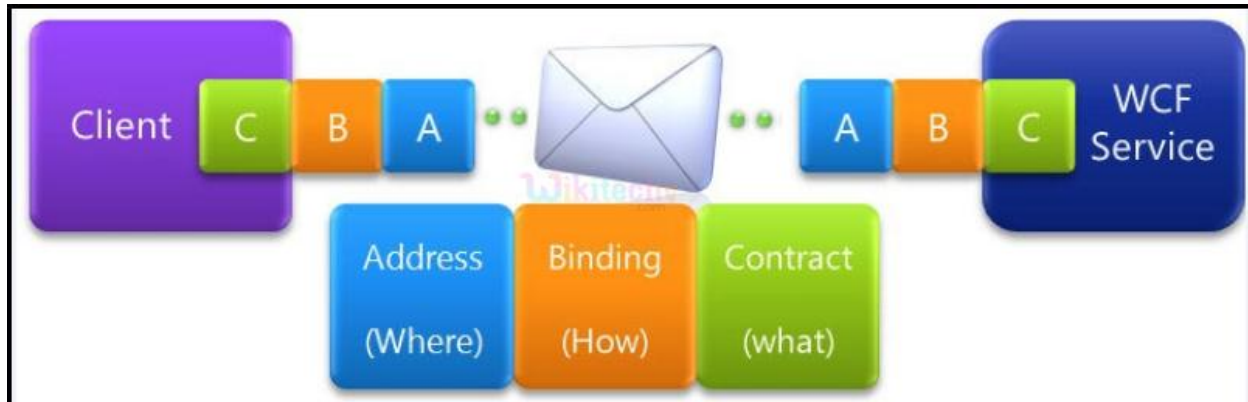


WCF-Service-Endpoint

- ◆ The **endpoint** is the fusion of the address, contract and binding.
- ◆ Every **endpoint** must have all three elements and the host exposes the **endpoint**.
- ◆ **WCF Service** is a program that exposes a collection of **Endpoints**.
- ◆ Endpoints helps the clients to communicate with a WCF Service.
- ◆ End point contains three components in service endpoint.



WCF Endpoint Components

Address:

- ◆ It denotes the exact **URL (Uniform Resource Locator)** address of the service to be used by a client for connecting to the service.
- ◆ A URL can be in the below form:

http://localhost:8732/Design_Time_Addresses/WikiTechy_WcfServiceLibrary/Service1

Binding:

Binding specifies how to communicate with the endpoint.

◆ It includes:

- ✎ Transport Protocol is ready to use. (Eg: TCP or HTTP).
- ✎ Encoding is used for the messages (Eg: text or binary).
- ✎ Security requirements is necessary. (Eg: SSL or SOAP message security).

Transport :

- ◆ It is the protocol for transmission like HTTP Named **Pipes, TCP, and MSMQ.**

Encoding (Optional):

- ◆ Encoding terminologies such as Plain Text, Binary or Message Transmission Optimization Mechanism (MTOM – for larger messages) are optional.



Protocol:

- Optional section is used to define security for the message.

The following table gives some list of protocols supported by WCF binding.

Binding	Description
BasicHttpBinding	Basic communication of web service with zero security.
WSHttpBinding	Added with WS-* support for transactions.
WSDualHttpBinding	Web service duplex contract with transactional support.
WSFederationHttpBinding	Web services including federated security along with transactional support.
MsmqIntegrationBinding	MSMQ application Communication with transactional support.
NetMsmqBinding	Applications queuing with transactional support.

NetNamedPipeBinding	Communication between applications on same IPAdrees with duplex contract and transaction support.
NetPeerTcpBinding	Communication between applications across peers with duplex contract support.
NetTcpBinding	Communication between applications across systems with duplex contract and transaction support.

Contract:

- ◆ It is a collection of operation provided by a **service endpoint**.
- ◆ The Contract **specifies** the Interface name for the client to know the exposed operations of a service.
- ◆ Every operation has one of the **exchange patterns, one-way, duplex and request/reply**.

App.config file code:



```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.web>
    <compilation debug="true" />
  </system.web>
  <!-- When deploying the service library project, the content of the config
file must be added to the host's
app.config file. System.Configuration does not support config files for
libraries. -->
  <system.serviceModel>
  <services>
```

```
<service name="WikiTechy_WcfServiceLibrary.Service1">
<host>
  <baseAddresses>
    <add baseAddress =
"http://localhost:8732/Design_Time_Addresses/WikiTechy_WcfServiceLibrar
y/Service1/" />
  </baseAddresses>
</host>
  <!-- Service Endpoints -->
  <!-- Unless fully qualified, address is relative to base address supplied
above -->
  <endpoint address="" binding="wsHttpBinding"
contract="WikiTechy_WcfServiceLibrary.IService1">
    <!--
      Upon deployment, the following identity element should be
      removed or replaced to reflect the
      identity under which the deployed service runs. If removed, WCF
      will infer an appropriate identity
      automatically.
    -->
    <identity>
      <dns value="localhost"/>
    </identity>
  </endpoint>
  <!-- Metadata Endpoints -->
  <!-- The Metadata Exchange endpoint is used by the service to
describe itself to clients. -->
  <!-- This endpoint does not use a secure binding and should be
secured or removed before deployment -->
  <endpoint address="mex" binding="mexHttpBinding"
```

```
contract="IMetadataExchange"/>
  </service>
</services>
<behaviors>
  <serviceBehaviors>
    <behavior>
      <!-- To avoid disclosing metadata information,
      set the value below to false and remove the metadata endpoint
      above before deployment -->
      <serviceMetadata httpGetEnabled="True"/>
      <!-- To receive exception details in faults for debugging purposes,
      set the value below to true. Set to false before deployment
      to avoid disclosing exception information -->
      <serviceDebug includeExceptionDetailInFaults="False" />
    </behavior>
  </serviceBehaviors>
</behaviors>
</system.serviceModel>
</configuration>
```



Explanation:



- 1 App.config file is automatically created in a **WCF Service Library application** to indicate the configurations for address, endpoints and bindings of a service.
- 2 and 3 The address attribute of the service denotes the **service address**. The element `<endpoint address="" binding="wsHttpBinding" contract="WikiTechy_WcfServiceLibrary.IService1">` defines the endpoint of the service that is exposed to clients.
- 4 `wsHttpBinding` represents an interoperable binding that supports distributed transactions to make it secure.
- 5 The contract attribute specifies the interface of the service as `WikiTechy_WcfServiceLibrary.IService1`.

- 6 The `<identity>` element permits the client to specify the probable service identity at run time. When the communication establishes between the client and service, WCF architecture will authenticate that the identity of the service matches the value of this element.
- 7 The line `<dns value="localhost"/>` specifies the domain address of the service. Here the service is in the same machine and hence the value is present as **"localhost"**.
- 8 The Metadata endpoint is used by the service to describe the operations via metadata to the clients via the line `<endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange"/>`.
- 9 MEX endpoints are a special way to allow clients to access the service's metadata via SOAP messages without using http get requests. This is denoted by `address="mex"`.
- 10 **mexHttpBinding** returns metadata for the client to build a proxy. This is enabled by the attribute `binding="mexHttpBinding"`.
- 11 The **IMetadataExchange Interface** provides method signature to return metadata about a service. This is implemented by the attribute `contract="IMetadataExchange"`.