* Project creation :-

→ WCF ⇒ create project in .NET framework
→ WCF Service Project

1) Create the service.
2) host the service. (In visual studio "WCF Service Host
   - Console application.
   - windows form application
   - " activation service.
3) test the service
4) 3) Use the service
   ↳ consume the service.
   client :- - browser
   - Desktop application
   - console application.

VS: ⇒ WCF test client ⇒ used to test the client

→ <ServiceModel> is assembly, has all thing which is required by ~~the~~ web service
(for hosting of the)

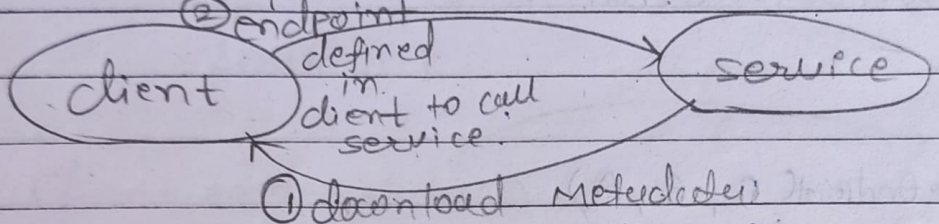
→ endpoint ⇒ ABC
Address ⇒ "" ⇒ means use default base address otherwise it be combine with base add^{n}.

using "get" request Metadata can be downloaded

→ HttpGetMexEnabled (upcoming class)

Http protocol → used to stop downding of Metadata by

↳ False :- Metadata can't be downloaded by the client, but service is running & if client has Metadata already downloaded, then client can use it got from the Metadata.



Message exchange bet^n client & server in SOAP Message.

→ Marshaling / UnMarshaling : } convert string
Serialization / Deserialization. } to int / int → str.

Y.P:- test the service before it is used by the client, hence problem can be trace.

→ Two App.config ⇒ ① for client
② for service } both are in same solution

→ For getting wsdl file add the service.

→ Service / Client ⇒ called as proxy class.
because that class is not Implemented in client, but in service, even though it's act like it itself containing it.

Page
Date

* 30-12-22 (dynamically link library)

→ Class library ⇒ dll file.
↳ Required when we create host of the service

Linux:- .so ⇒ ~~source object~~ share object
↳ to create library.

For we library: System.serviceModel
this 2 are → WCFService.dll
needed.

→ AddServiceEndPoint(C, B, A)
Reverse
↳ while specifying service host we can add base address.

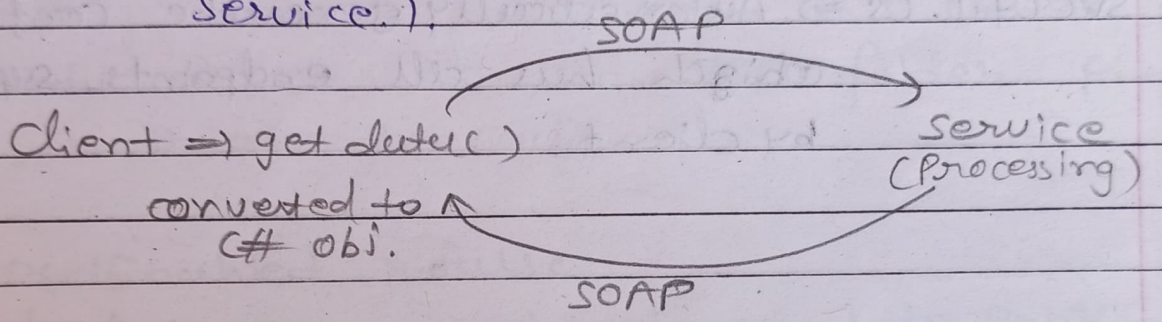
→ typeof (ServiceClass) : actual class is not used it's type is used at runtime from dll file it will be combined.

→ For every service
↙ ↘
Service Behaviour

search { Tcp 3way hand connection
Tcp state diagram
0.0.0.0:8000 Vs 0::0:8000

→ On client side ⇒ only application's endpoint is required, because client is not exposing Metadata.

→ Proxy class ⇒ Method (Implementation) not in client, but we are calling it like it is in client. (Implementation is in service.).



↳ while creating Proxy class object, specify endpoint

→ If service, host client are in different so it then you need to run service (i.e. host) then only client will able to download Metadata.

⇒ Named pipe ⇒ It will create file, hence it (data) will be stored even after terminate / exit the process.

Q. Pipe :- can be used only in same machine as it is used for IPC.

→ basicHTTPBinding ⇒ use 1 version.
↳ Not interoperable
↳ work for older standard.

Page
Date

4.f :- Multiple Protocol should be supported by the host as All client will not use always same Protocol.

→ Namespace should be specified to Identify end service uniquely.

→ svcutil.cs ⇒ Automatically generate config which which has all endpoints supported by client.

★ 11-1-23 :-

→ service contract :- Interface (which services are available)
→ If we have custome types, then we need to have data contract.
(whether data should be in body / header etc)
→ Message contract : (control)
↳ Provide more flexibility to message format.
↳ otherwise Normally it's in SOAP.

4.g :- WCF is only supported by .NET framework, can't by .NET Core

→ [DataMember] :- To able to convert to string.
↳ enable the serialization.

→ [DataContract] :- required for custom class.

→ Create the service
Create the host

→ Pipe :- why Port no. is not required?
↳ service is running in the same system & it is named pipe hence it's unique within the system (i.e. In same folder pipe path is unique for each.

→ HttpGetEnabled = false
↳ As we are not using http Protocol for downloading metadata.

Y.F:- In http only ⇒ localhost take bydefault port 80

→ Two Processes are called related If they have same ancestor.
{ Pipe, fork() } To be shared pipe between children
}
fork() } can't be shared.
Pipe

→ Only one endpoint is required to download the metadata, not both.

→ If there are multiple endpoint, then you need

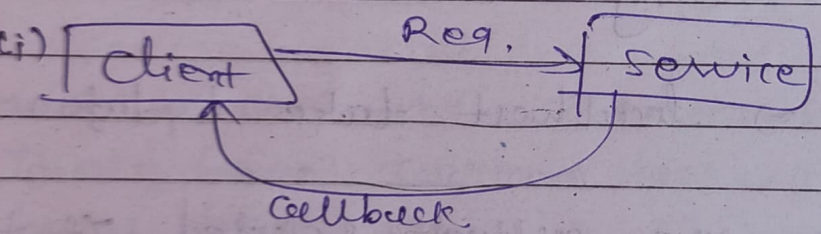
to explicit specify the endpoint Name.
↳ namespace.ServiceBusClient Variable.

* Message Contract :-

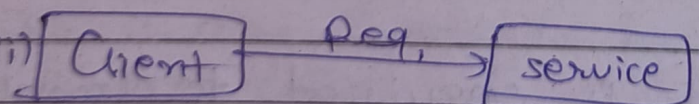
→ An Intermmediate node can Process the header, can't body.

* 12-1-23 :-

* Message Exchange patterns :-

→ (i) 

↳ In production environment, we can't get stack trace If error occured, hence using callback from service again calling client for stack trace.

(ii) 

} Client is not expecting any res.

↳ one way Message : when you are not expecting the Response.
↳ Ex :- Sleep op^n.

→ 8733 & Test using TCP & Pipe

Page
Date

Q.P :→ Build cpp. sometimes need to be run as administrator as some port are restricted by the system.

→ csc / wsdl : It will download the wsdl file, which is visible in the browser.

(ii) One-way

→ Hence, it's like Asynchronous code as it is not waiting for complete execution of service function.

Q.P :- In request-reply, when req. is done for first time it will take time as it's is establishing connection bet^n client & server.

(iii) Req-callback.

1) executed on serverside
2) executed on clientside.
   ↳ In client you need remember what it has done in client.

Property :- callback Contract = type of (Interface name) which need to (Suppose to be) ↓ Implement by client.

1) Operation Context

2) Instance Context
   ↳ Store the service of object.