**Introduction:**

|- History of the RESTful Web Services

|-Purpose of RESTful Web Services

|-Principles of RESTful Services (Adopted from Principles of Web)

|-1. Unique Addressable URI (URI=Unique Resource Identifier)

|-2. Uniform, Constrained Interfaces (it is not java Interface)

|-Advantages of Uniform, Constrained Interfaces

|-1. Familiarity

|-2. Interoperability

|-3. Scalability

|-3. Representation-Oriented

|-4. Communicate Statelessly

|-Advantages

|-Avoids the Thread-Starvation Issues

|-5. HATEOAS (Hypermedia As The Engine Of Application State)

|-Use-Case Related to HATEOAS

|-Advantages of HATEOAS

|-1. The engine of application state

|-2. Avoids/Eliminates (WSDL) the Service Description Documentation

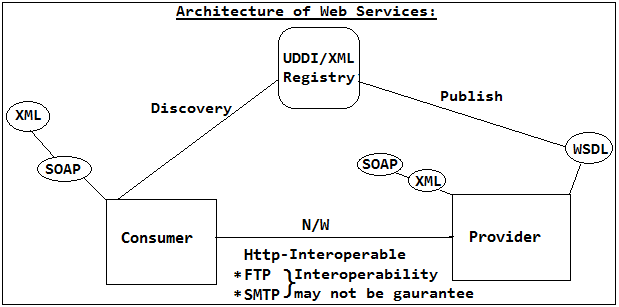
|-Architecture of RESTful Services

**Architecture of RESTful Services (Representational State Transfer Services):**

What is RESTful service can u plz explain?

RESTful service are one form of the Web Services that means the new architectural style of developing the Web services is called as RESTful services.

The Web Services are bounded with set of rules and standards unless until we bound to the standards that provided by the WS-I we cannot dev the web services. That means the programming is surrounded with standards and protocol specifications but when it comes to the RESTful services there is no standards in developing the RESTful hence there is no WS-I organization for the RESTful services rather it is kind of Architectural style over which we can develop the applications that is the reason it is called as new architectural style of developing the Web services.



Provider is seems to be a restricted service which will expects the data in an specific format surrounded with standards and protocol specifications but RESTful accessed in an Uniform (easy/simple) way as like Web hence we should not call it as Provider (Service) rather we need to called as Resources. Hence we can refer the services that we are trying to develop using REST is called as REST-Service/Resource/Web-Resource bcz these can accessed like a web resource just by using single URI hence we will name Consumer as Client.

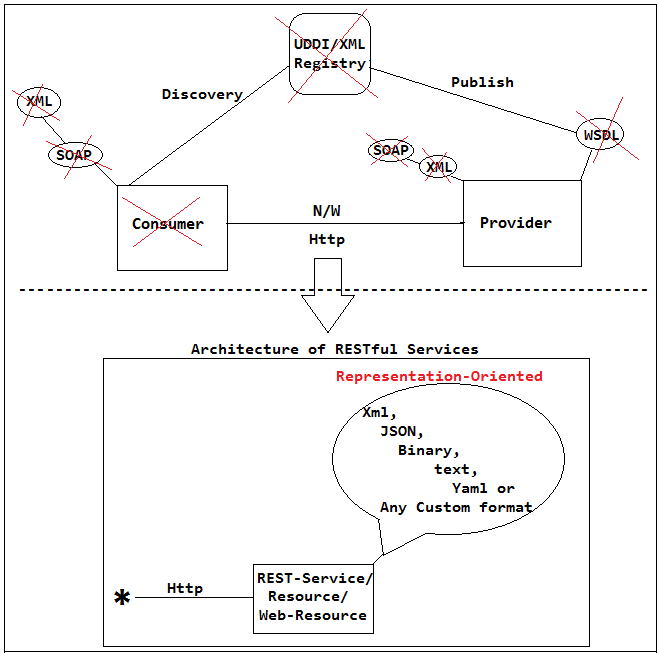
RESTful is not requires any data formats or SOAP protocols hence in the architecture we will completely eliminate the SOAP and MEF.

Resource can accept xml, JSON, binary, text, Yaml or any custom format etc that we wish to access the resource that means the resource not only serve the one data format rather it can serve of the data representational that we are looking for which is called Representational Format/Representation-Oriented.

We can any format but recommended to Xml, JSON etc. But it not recommended to use binary bcz interoperability will not be guaranteed.

If client wants to access the Resource then we can use any protocol in RESTful but JAX-RS API has been designed based on the Http protocol only hence we need to use only Http only to access the Resources.

If we client wants to access the Resource description language is not required like WSDL hence we need to discover or we no need to publish hence we don't need UDDI-registry in case of RESTful. Hence we can access the Resource by simply sending the req as like web hence who will be cable of sending the Http req can acts as Client hence there is development of the Client (Consumer or Stub) as well which is called as "New Architectural Style of Developing the Web Services" which is called as "RESTful services".



🡪What is the RESTful services and how does the RESTful services different form Web Services?

Ans: Explain the Architecture