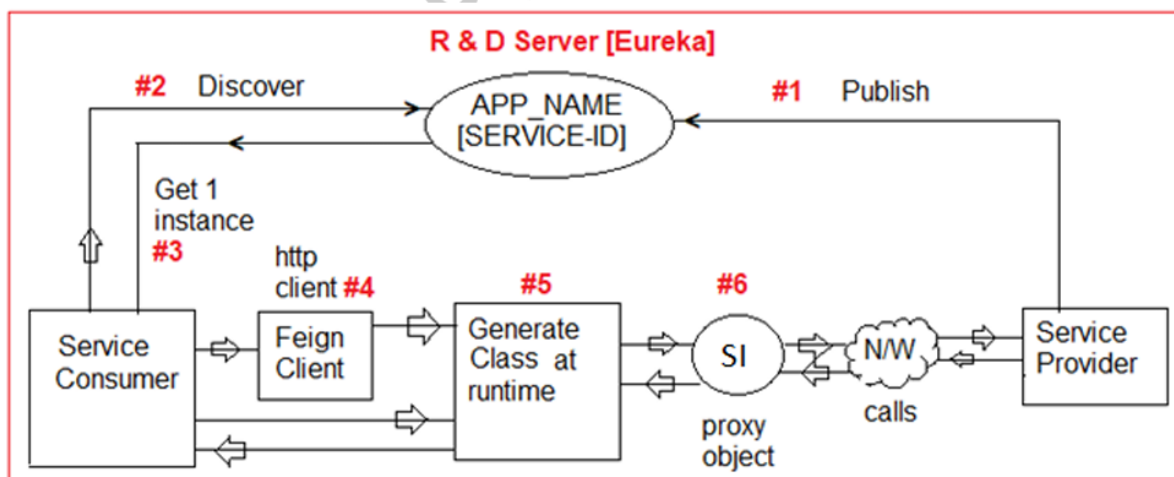**Feign Client:**

Feign is Declarative Client, which supports generating code at runtime and proxy Objects by using Proxy HTTP Request calls can be made. =>It supports even Parameters (Path/Query…) and Global Data Conversion (XML/JSON).

Work Flow Design: --

1. Interface defined with @FeignClient validates all details (ServiceId, Paths, Method Types…).

2. Uses ServiceId and communicates with R&D it gets LBSR (Load Balancing).

3. A Proxy class is generated which makes all this process. Based on load Factor one Service Instance is chosen. ***Ribbon is used internally.

4. Http Logic is generated using Service Instance (URI).

5. Proxy class supports making Http Request and Response.

Work Flow Design:--

=>Feign Client is an Interface and contains abstraction details, it takes I/P details from programmer like:

a>path (Provider Paths at class and method).

b>Http Method Type (GET, POST…).

c>ServiceId (Application Name).

d>Input Details and Output Type (String, Object, Collection).

e>Parameters (Request, Path,).

=>We need to apply Annotation at starter class level @EnableFeignClients.

=>At interface level apply @FeignClient (name=" serviecId").