Overview of Netflix components-

Spring Cloud Netflix provides Netflix OSS integrations for Spring Boot apps through autoconfiguration and binding to the Spring Environment and other Spring programming model idioms. With a few simple annotations you can quickly enable and configure the common patterns inside your application and build large distributed systems with battle-tested Netflix components. The patterns provided include Service Discovery (Eureka), Circuit Breaker (Hystrix), Intelligent Routing (Zuul) and Client Side Load Balancing (Ribbon).

|  |  |
| --- | --- |
| **Netflix Component Name** | **Functionality** |
| Eureka | Service Registration and Discovery |
| Ribbon | Dynamic Routing and Load Balancer |
| Hystrix | Circuit Breaker |
| Zuul | Edge Server |

What is Netflix Hystrix? Need for it?

**Hystrix is a latency and fault tolerance library designed to isolate points of access to remote systems, services and 3rd party libraries, stop cascading failure and enable resilience in complex distributed systems where failure is inevitable.**  
Usually for systems developed using Microservices architecture, there are many microservices involved. These microservices collaborate with each other.  
Consider the following microservices-

* Fallback method
* Circuit Breaker

What is the Netflix Zuul? Need for it?

Zuul is a JVM based router and server side load balancer by Netflix.   
It provides a single entry to our system, which allows a browser, mobile app, or other user interface to consume services from multiple hosts without managing cross-origin resource sharing (CORS) and authentication for each one. We can integrate Zuul with other Netflix projects like Hystrix for fault tolerance and Eureka for service discovery, or use it to manage routing rules, filters, and load balancing across your system.

**EUREKA**

Get ZUUL

**Student-Service**

**School-Service**

Instances from

**Z**

**U**

**U**

**L**

**G**

**A**

**T**

**E**

Eureka 1 Register Zuul Gateway to Eureka

2 3

**Calling Zuul**

**Gateway**

**to fetch Service**