Diagram 1

```
                    ┌──────────────┐
                    │ JWT SECURITY │
                    └──────────────┘
                         ↑  ↓
┌──────────┐        ┌─────────────┐        ┌───────────────┐        ┌──────────────┐
│ END USER │───────→│ API GATEWAY │───────→│ EUREKA SERVER │───────→│ MICROSERVICE 1│
└──────────┘        └─────────────┘        └───────────────┘   │    └──────────────┘
                                                               ├───→┌──────────────┐
                                                               │    │ MICROSERVICES│
                                                               │    └──────────────┘
                                                               └───→┌──────────────┐
                                                                    │ MICROSERVICE2│
                                                                    └──────────────┘
                                               ┌───────────────┐
                                               │ CONFIG SERVER │
                                               └───────────────┘
```

Diagram 2

```
┌──────────┐      ┌─────────────┐                                  ┌──────────────────────┐
│ END USER │─────→│ API GATEWAY │                               ┌─→│ CREATE JWT REQUEST    │
└──────────┘      └─────────────┘                               │  │ OBJECT                │
                         ↓                                      │  └──────────────────────┘
                  ┌─────────────┐       ┌──────────────────┐    │        ↓          ↓
                  │ PREFILTER   │       │ HTTP REQUEST READ │   │              ┌──────────────┐
                  └─────────────┘       │ USERNAME AND      │   │              │ WRONG        │
                         ↓              │ PASSWORD BY       │──→│              │ CREDENTIAL   │
┌────────────────────────────┐         │ READING REQUSET   │                  │ WILL GIVE    │
│ RUN METHOD                 │         │ QUERY             │                  │ EXCEPTION    │
│ Add logic to integrate     │         └──────────────────┘                  └──────────────┘
│ jwt security to api        │                                    ┌──────────────────────────┐
│ gateway- Interservice      │                                    │ CREATE HTTP HEADER        │
│ communication- REST        │                                    │ TYPE JSON                 │
│ TEMPLATE bean in           │                                    │ HTTP ENTITY WILL          │
│ main method                │                                    │ CARRY JWT REQUEST         │
└────────────────────────────┘                                    │ AND HEADER                │
                                                                   └──────────────────────────┘
┌────────────────────┐                                                       ↓
│ can't access if not │                        ┌──────────────────────────────────┐
│ authorized          │                        │ REST TEMPLATE WILL                │
└────────────────────┘                        │ COMMUNICATE WITH URL AND           │
                                               │ CREATE AUTH RESPONSE               │
┌──────────────────────────┐  ┌─────────────┐ └──────────────────────────────────┘
│ Redirect to requested     │  │ Verification │              ↓
│ end point if authorized   │  │ create new   │ ┌─────────────────────────────────────┐
└──────────────────────────┘  │ haeder with  │ │ PARSE STRING METHOD CONVERTS STRING  │
                              │ get method-  │ │ TO OBJECT, USE OF JSON FLATTNER AND   │
                              │ type json-   │ │ RETURNS TOKEN                        │
                              │ using        │ └─────────────────────────────────────┘
                              │ authozation +│
                              │ bearer+ token│
                              └─────────────┘
```

1. User send request with credentials
2. Json web token is returned(if user wants to access protected endpoint, user should sent jwt in authorization header using bearer
3. Synchronous communication is the request and response in form of json. It waits for response, (BLOCKING CALL) can not proceed further without response.
4. Here gateway is sending request and jwt security is sending response, based on response, gateway will go to next step
5. Gateway uses round robin technique(load balancing technique) (round robin is the simplest method that routes request across all the instances sequentially, let say if we have two ports 7070 and 7072. It will first go to 7070 and then to 7072 and keep repeating.
6. Rest template - api to api call happens
7. Json flattner and json smart dependency — we need to parse string to json object