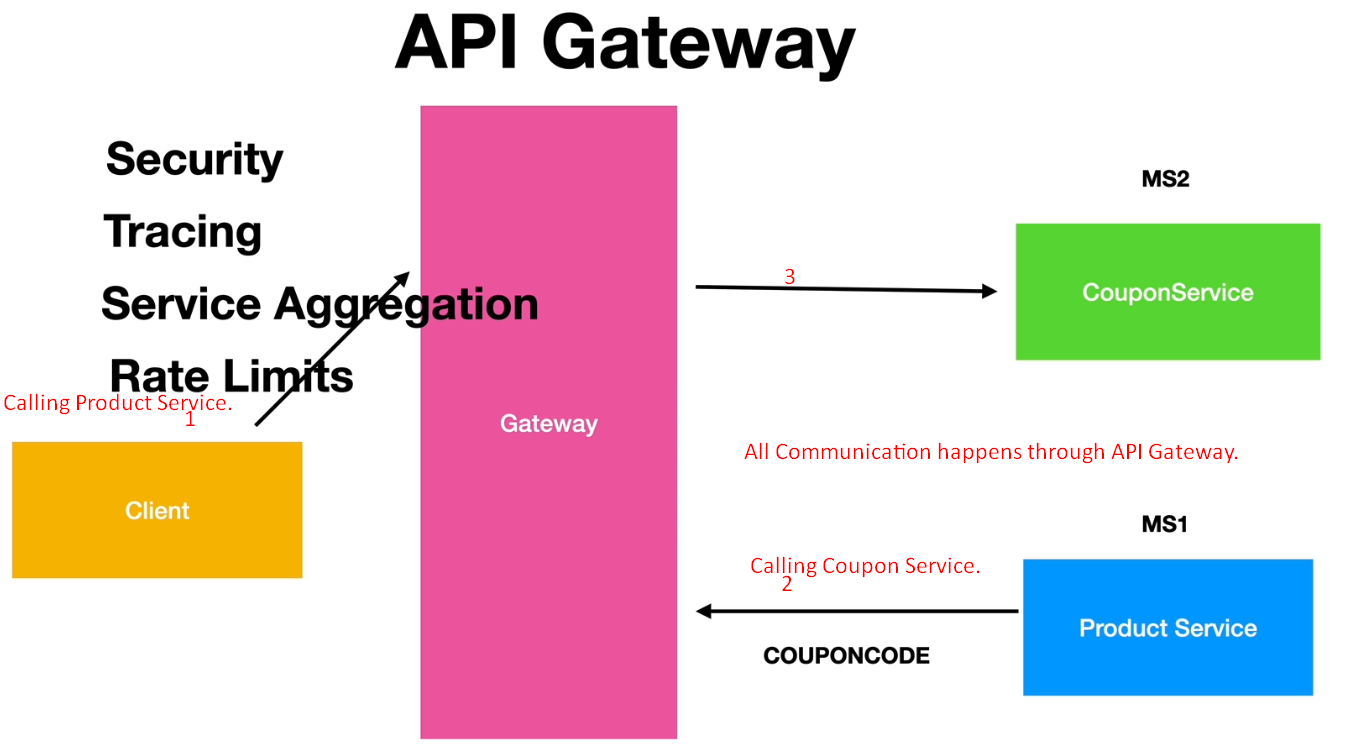
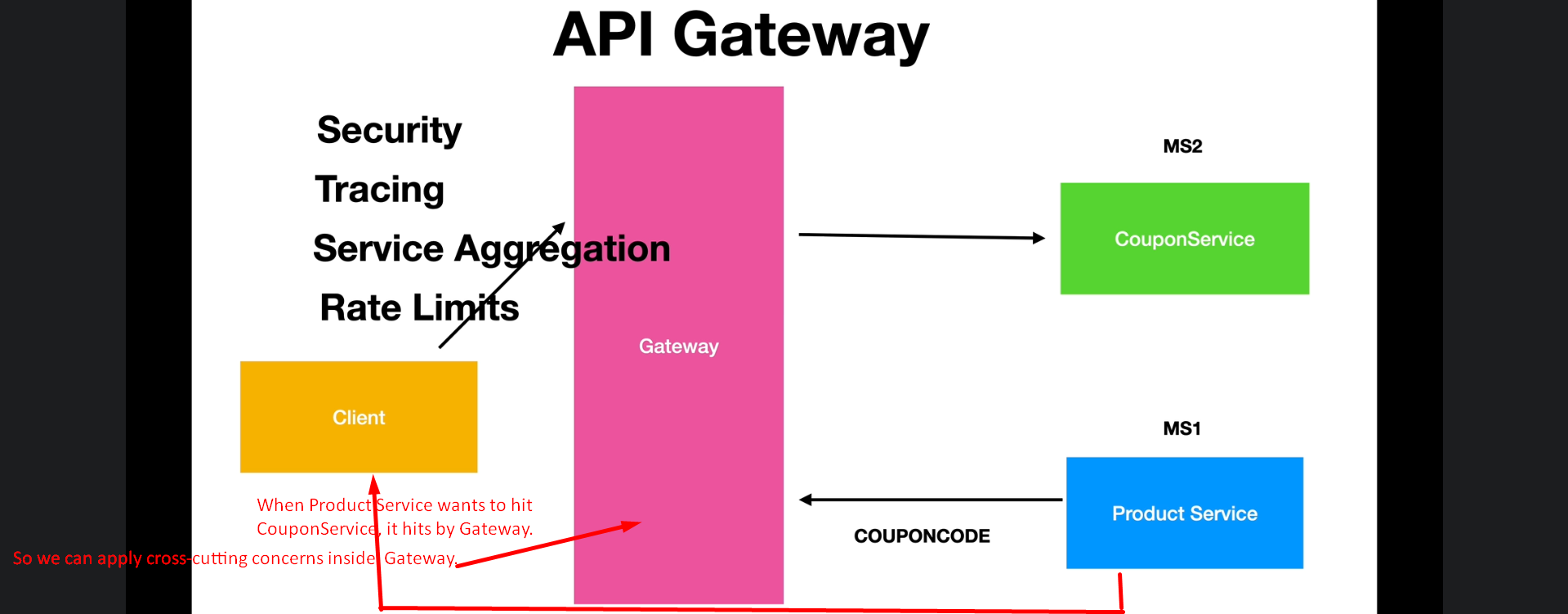
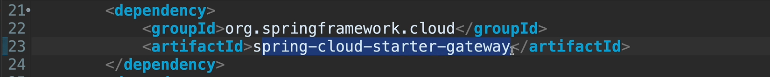
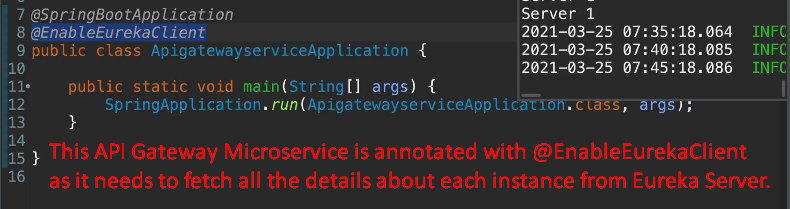
1. We have **non-functional requirements** across microservices; this could be security like authentication.
   1. This could be security like authentication, authorization, encryption, decryption, tracing.
   2. Tracing where we need to track the incoming request as it flows through microservices.
   3. **Service Aggregation**: Where a client needs to make calls to various microservices.
   4. **Rate Limits**: When you’re using cloud like AWS or Microsoft Azure you want to control the resources that each of your microservice can use.  
      These all are called **Cross Cutting Concerns**.  
      So instead of repeating these crosscutting concern implementations across microservices, we use Spring Cloud API Gateway.
2. **Spring Cloud API Gateway**:
   1. Cross-cutting concerns are configured into API Gateway Project.
   2. All the requests are redirected to this Gateway.



1. 
   1. We write cross-cutting concerns in Gateway API Project.
   2. All the microservices are redirected through the API Gateway.
2. Steps:
   1. Include Dependency:  
      
   2. 
   3. Configuration for Routes for each microservice in application.properties.  
      **NOTE**: Same configuration can be done with java too.  
      