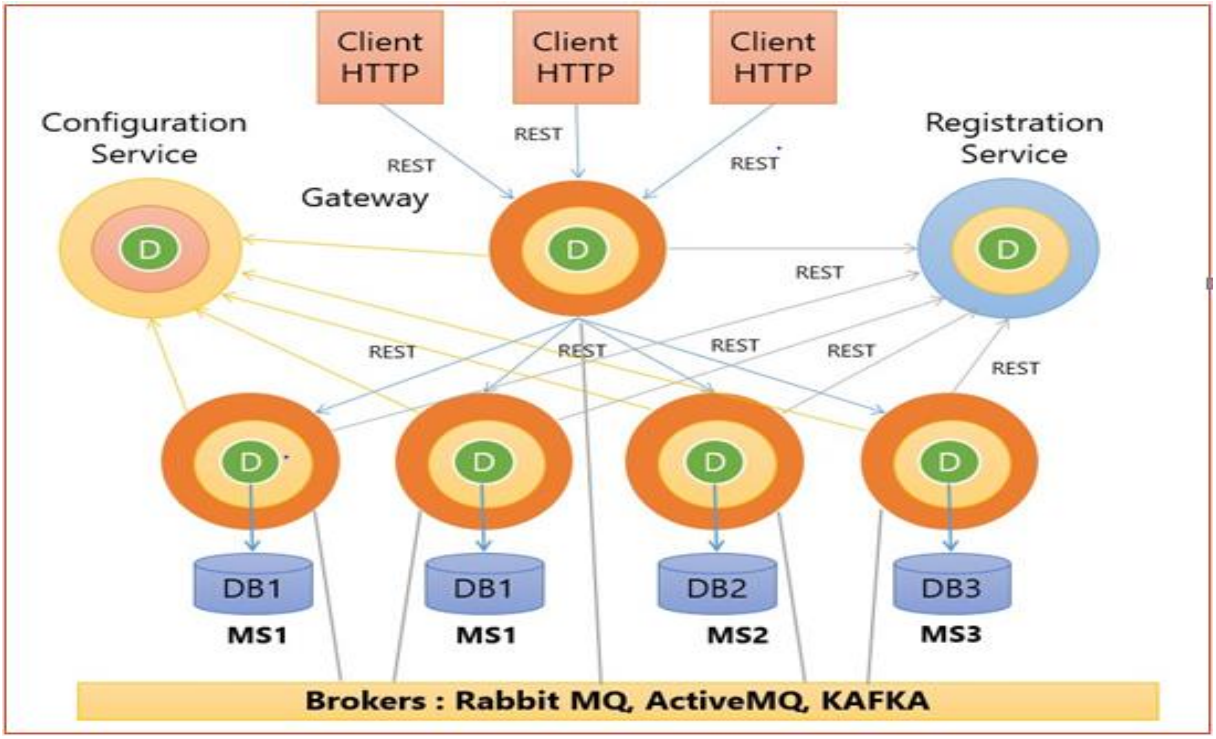
ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

Rapport Architectures Micro Services

KSOURI Elmehdi
EMSI CENTRE G2

I. INTRODUCTION :



Les micro-services sont une approche d'architecture et de
Développement d'une **application composées de petits services**.

L'idée étant de découper un grand problème en petites unités
Implémentée sous forme de micro-services

Chaque service est responsable d'une fonctionnalité,

Chaque micro-service est **développé**, **testé** et **déployé**
Séparément des autres.

Chaque micro service est développé en utilisant une technologie quipeut être différ

des autres. (java,c++,python,.NET)

- Dans ce TP, on va voir comment on peut créer des web servi

  distribués avec le Spring Cloud en utilisant Eureka et Discove

  service.

II. Code Source :

https://github.com/Mehdiksouri/Architecture-Micro-service

III. Capture d'écran :

```java
@Data
@Entity
@AllArgsConstructor
@NoArgsConstructor
@ToString
public class Customer {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private String email;
}
```

Ce micro-service prend en charge toutes les opérations relatives aux données client, il utilise sa propre base de donné

qu'il s'enregistre dans le service d'enregistrement eurekaDiscovery-service : spring.cloud.discovery.enabled = true

```java
@FeignClient(name = "CUSTOMER-SERVICE")
public interface CustomerRestClient {
    @GetMapping("/customers/{id}")
    Customer getCustomerById(@PathVariable(name="id") Long id);
}
```

Alors le @FeignClient permet d'établir une communication vers un autre micro service qui est enregistré dans le discov

service.

```java
@CrossOrigin("*")
@RestController
public class BillRestController {

    private BillRepository billRepository;
    private ProductItemRepository productItemRepository;
    private CustomerRestClient customerRestClient;
    private ProductItemRestClient productItemRestClient;

    public BillRestController(BillRepository billRepository,
ProductItemRepository productItemRepository, CustomerRestClient
customerRestClient, ProductItemRestClient productItemRestClient) {
        this.billRepository = billRepository;
        this.productItemRepository = productItemRepository;
        this.customerRestClient = customerRestClient;
        this.productItemRestClient = productItemRestClient;
    }
```

Et le @RestController pour un contrôleur REST et @CrossOrigin pour que l'application front-end puisse consommer

l'api REST

```java
@SpringBootApplication
public class GatewayApplication {

    public static void main(String[] args) {
        SpringApplication.run(GatewayApplication.class, args);
    }


    //@Bean
    RouteLocator routeLocator(RouteLocatorBuilder routeLocatorBuilder){
        return routeLocatorBuilder.routes()
                .route((r)->r.path("/customers/**").uri("lb://CUSTOMER-
SERVICE"))
                .route((r)->r.path("/products/**").uri("lb://INVENTORY-
SERVICE")).build();
    }

    @Bean
    DiscoveryClientRouteDefinitionLocator
definitionLocator(ReactiveDiscoveryClient client,

DiscoveryLocatorProperties properties){
        return new DiscoveryClientRouteDefinitionLocator(client,
properties);
    }
```

Pour le gateway on peut le configurer d'une manière statique (on donne les uris des micro services) ou même dynamiq

laisse l'application d'avoir les services en se basant sur le nom de chaque service).

```java
@SpringBootApplication
@EnableEurekaServer
public class EurekaDiscoveryApplication {

    public static void main(String[] args) {
        SpringApplication.run(EurekaDiscoveryApplication.class, args);
    }

}
```

On active le mode eureka server pour que l'application soit un service de registration.

KAFKA

C:\WINDOWS\system32\cmd.exe - bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic R1

```
        test
{"name":"test","user":"U1","date":"2022-01-13T22:04:34.223+00:00","duration":2862}
{"name":"blog","user":"U1","date":"2022-01-13T22:05:06.951+00:00","duration":6662}
{"name":"blog","user":"U2","date":"2022-01-13T22:07:38.906+00:00","duration":3276}
{"name":"blog","user":"U2","date":"2022-01-13T22:07:39.880+00:00","duration":8719}
{"name":"blog","user":"U1","date":"2022-01-13T22:07:40.596+00:00","duration":1001}
{"name":"blog","user":"U2","date":"2022-01-13T22:07:40.784+00:00","duration":7605}
```

C:\WINDOWS\system32\cmd.exe - bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic R1

```
        test
{"name":"test","user":"U1","date":"2022-01-13T22:04:34.223+00:00","duration":2862}
{"name":"blog","user":"U1","date":"2022-01-13T22:05:06.951+00:00","duration":6662}
{"name":"blog","user":"U2","date":"2022-01-13T22:07:38.906+00:00","duration":3276}
{"name":"blog","user":"U2","date":"2022-01-13T22:07:39.880+00:00","duration":8719}
{"name":"blog","user":"U1","date":"2022-01-13T22:07:40.596+00:00","duration":1001}
{"name":"blog","user":"U2","date":"2022-01-13T22:07:40.784+00:00","duration":7605}
{"name":"blog","user":"U2","date":"2022-01-13T22:41:19.774+00:00","duration":521}
{"name":"blog","user":"U2","date":"2022-01-13T22:42:02.337+00:00","duration":4033}
{"name":"test"}
```

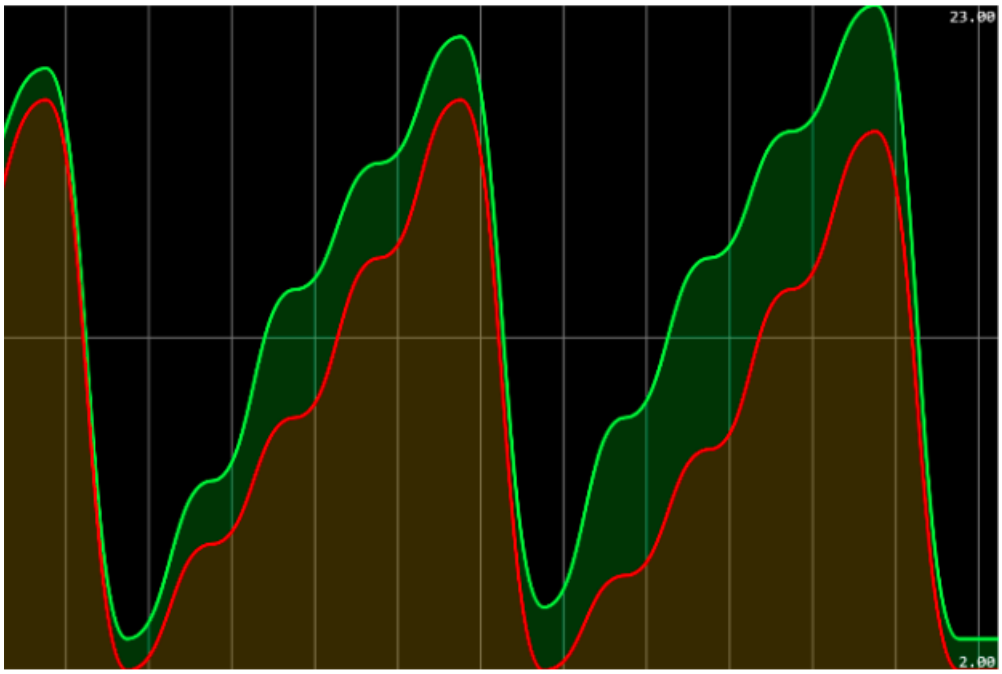
C:\WINDOWS\system32\cmd.exe - bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic R2

```
{"name":"P1","user":"U1","date":"2022-01-16T14:48:19.840+00:00","duration":3728}
{"name":"P2","user":"U1","date":"2022-01-16T14:48:29.853+00:00","duration":2754}
{"name":"P2","user":"U2","date":"2022-01-16T14:48:39.855+00:00","duration":7512}
{"name":"P1","user":"U1","date":"2022-01-16T14:48:49.866+00:00","duration":5931}
{"name":"P2","user":"U1","date":"2022-01-16T14:48:59.876+00:00","duration":853}
{"name":"P1","user":"U1","date":"2022-01-16T14:49:09.889+00:00","duration":350}
{"name":"P2","user":"U2","date":"2022-01-16T14:49:19.896+00:00","duration":4491}
{"name":"P2","user":"U1","date":"2022-01-16T14:49:29.912+00:00","duration":4675}
{"name":"P2","user":"U2","date":"2022-01-16T14:49:39.914+00:00","duration":1199}
{"name":"P1","user":"U1","date":"2022-01-16T14:49:49.928+00:00","duration":7314}
{"name":"P1","user":"U2","date":"2022-01-16T14:49:59.936+00:00","duration":4425}
{"name":"P2","user":"U2","date":"2022-01-16T14:50:09.944+00:00","duration":3928}
{"name":"P2","user":"U1","date":"2022-01-16T14:50:19.958+00:00","duration":1227}
{"name":"P1","user":"U2","date":"2022-01-16T14:50:29.969+00:00","duration":8297}
{"name":"P2","user":"U1","date":"2022-01-16T14:50:39.972+00:00","duration":5813}
{"name":"P2","user":"U2","date":"2022-01-16T14:50:49.980+00:00","duration":1782}
{"name":"P1","user":"U2","date":"2022-01-16T14:50:59.990+00:00","duration":8542}
{"name":"P2","user":"U2","date":"2022-01-16T14:51:10.001+00:00","duration":4382}
{"name":"P2","user":"U2","date":"2022-01-16T14:51:20.014+00:00","duration":423}
{"name":"P1","user":"U2","date":"2022-01-16T14:51:30.027+00:00","duration":8625}
{"name":"P1","user":"U1","date":"2022-01-16T14:51:40.040+00:00","duration":8362}
{"name":"P1","user":"U2","date":"2022-01-16T14:51:50.051+00:00","duration":1110}
{"name":"P1","user":"U2","date":"2022-01-16T14:52:00.066+00:00","duration":1705}
{"name":"P1","user":"U1","date":"2022-01-16T14:52:10.073+00:00","duration":1334}
{"name":"P1","user":"U2","date":"2022-01-16T14:52:20.076+00:00","duration":4180}
{"name":"P1","user":"U2","date":"2022-01-16T14:52:30.085+00:00","duration":5893}
{"name":"P2","user":"U2","date":"2022-01-16T14:52:40.090+00:00","duration":8621}
{"name":"P2","user":"U2","date":"2022-01-16T14:52:50.099+00:00","duration":2709}
{"name":"P1","user":"U2","date":"2022-01-16T14:53:00.111+00:00","duration":7257}
```

C:\WINDOWS\system32\cmd.exe - bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:90

```
=>2022-01-21T21:21:00Z2022-01-21T21:21:05ZP2    6
=>2022-01-21T21:21:00Z2022-01-21T21:21:05ZP2    9
=>2022-01-21T21:21:00Z2022-01-21T21:21:05ZP1    7
=>2022-01-21T21:21:00Z2022-01-21T21:21:05ZP2    13
=>2022-01-21T21:21:00Z2022-01-21T21:21:05ZP1    12
=>2022-01-21T21:21:00Z2022-01-21T21:21:05ZP2    16
=>2022-01-21T21:21:00Z2022-01-21T21:21:05ZP1    18
=>2022-01-21T21:21:00Z2022-01-21T21:21:05ZP1    24
=>2022-01-21T21:21:05Z2022-01-21T21:21:10ZP1    1
=>2022-01-21T21:21:05Z2022-01-21T21:21:10ZP2    2
=>2022-01-21T21:21:05Z2022-01-21T21:21:10ZP1    4
=>2022-01-21T21:21:05Z2022-01-21T21:21:10ZP2    8
=>2022-01-21T21:21:05Z2022-01-21T21:21:10ZP2    13
=>2022-01-21T21:21:05Z2022-01-21T21:21:10ZP1    8
=>2022-01-21T21:21:05Z2022-01-21T21:21:10ZP2    19
=>2022-01-21T21:21:05Z2022-01-21T21:21:10ZP1    11
=>2022-01-21T21:21:05Z2022-01-21T21:21:10ZP2    24
=>2022-01-21T21:21:05Z2022-01-21T21:21:10ZP1    15
=>2022-01-21T21:21:05Z2022-01-21T21:21:10ZP1    16
=>2022-01-21T21:21:05Z2022-01-21T21:21:10ZP2    29
=>2022-01-21T21:21:10Z2022-01-21T21:21:15ZP1    2
=>2022-01-21T21:21:10Z2022-01-21T21:21:15ZP2    2
=>2022-01-21T21:21:10Z2022-01-21T21:21:15ZP2    6
=>2022-01-21T21:21:10Z2022-01-21T21:21:15ZP1    8
=>2022-01-21T21:21:10Z2022-01-21T21:21:15ZP2    7
=>2022-01-21T21:21:10Z2022-01-21T21:21:15ZP1    17
```

localhost:8080



ANGULAR TS & NGRX