|  |  |
| --- | --- |
| Nama | : Rasihan Yusri Mazid |
| NIM | : 2301081011 |

**1. Server-Side Discovery**

Server-side discovery adalah metode di mana layanan tidak bertanggung jawab untuk menemukan layanan lain secara langsung. Sebagai gantinya, ketika sebuah layanan klien ingin mengakses layanan lain, ia mengirimkan permintaan ke load balancer atau service proxy, yang bertindak sebagai perantara. Load balancer ini akan berkonsultasi dengan service registry (pangkalan data layanan yang tersedia) untuk menemukan instance yang sesuai dan meneruskan permintaan ke layanan yang dituju.

**Cara kerja server-side discovery:**

1. Layanan terdaftar di service registry saat startup.
2. Klien mengirimkan permintaan ke load balancer/service proxy.
3. Load balancer mencari instance layanan yang tersedia di service registry.
4. Load balancer meneruskan permintaan ke instance layanan yang dipilih.

**Contoh implementasi:**

* Kubernetes Service (menggunakan kube-proxy)
* Netflix Ribbon + Eureka + Zuul

**2. Client-Side Discovery**

Client-side discovery adalah metode di mana klien sendiri yang bertanggung jawab untuk menemukan lokasi layanan tujuan dengan langsung berkomunikasi dengan service registry. Klien akan mengambil daftar instance layanan dari service registry, memilih salah satu berdasarkan algoritma load balancing, lalu mengirimkan permintaan langsung ke instance layanan tersebut.

**Cara kerja client-side discovery:**

1. Layanan terdaftar di service registry saat startup.
2. Klien langsung menghubungi service registry untuk mendapatkan daftar instance layanan yang tersedia.
3. Klien memilih instance layanan yang akan diakses berdasarkan load balancing.
4. Klien mengirimkan permintaan langsung ke instance layanan yang dipilih.

**Contoh implementasi:**

* Netflix Eureka + Ribbon
* Consul Client-Side Load Balancing

**3. Kelebihan dan Kekurangan**

|  |  |  |
| --- | --- | --- |
| **Metode** | **Kelebihan** | **Kekurangan** |
| Server-Side Discovery | - Klien lebih sederhana karena tidak perlu menangani service discovery.  - Load balancing lebih fleksibel karena dikelola secara terpusat.  - Cocok untuk sistem dengan banyak layanan dinamis. | - Membutuhkan load balancer/service proxy, yang bisa menjadi bottleneck.  - Infrastruktur lebih kompleks karena ada komponen tambahan. |
| Client-Side Discovery | - Tidak memerlukan load balancer eksternal, sehingga mengurangi overhead.  - Lebih cepat karena klien langsung memilih instance layanan. | - Klien harus mengimplementasikan service discovery dan load balancing sendiri.  - Jika klien memiliki banyak layanan, kompleksitas di sisi klien meningkat. |

**4. Service Registry**

Service registry adalah komponen yang bertindak sebagai direktori layanan dalam arsitektur microservices. Registry ini menyimpan daftar layanan yang tersedia beserta informasi seperti alamat IP, port, dan status kesehatan layanan.

**Fungsi utama service registry:**

1. Menyediakan daftar layanan yang tersedia.
2. Memantau kesehatan layanan dan menghapus layanan yang tidak aktif.
3. Mengizinkan layanan baru untuk mendaftarkan diri saat startup.

**Contoh implementasi service registry:**

* Netflix Eureka
* Consul
* Apache Zookeeper
* Kubernetes Service Discovery

Service registry digunakan dalam server-side discovery dan client-side discovery untuk memastikan bahwa layanan dapat saling menemukan dengan cara yang efisien.