

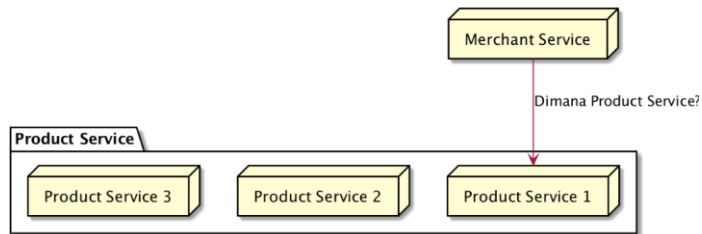
## 16 Server Side Discovery

Server-side discovery adalah pola desain dalam sistem microservices di mana tanggung jawab untuk menemukan lokasi layanan (service discovery) diletakkan pada server atau komponen infrastruktur daripada pada klien. Ini kontras dengan client-side discovery, di mana klien bertanggung jawab untuk menentukan lokasi layanan yang ingin diakses.

Karakteristik Server-Side Discovery:

- **Service Registry:**  
Pusat Penyimpanan Informasi Layanan: Service registry adalah komponen pusat yang menyimpan informasi tentang lokasi dan status layanan yang tersedia. Layanan mendaftar ke registry saat mereka beroperasi dan dihapus dari registry saat mereka berhenti atau crash.  
Dinamis: Informasi dalam registry diperbarui secara dinamis sesuai dengan perubahan status layanan.
- **Load Balancer atau Service Proxy:**  
Load Balancer: Mengarahkan permintaan dari klien ke instance layanan yang sesuai berdasarkan informasi dari service registry. Contoh: AWS Elastic Load Balancer, Google Cloud Load Balancer.  
Service Proxy: Bertindak sebagai perantara antara klien dan layanan, mengambil informasi dari service registry untuk menemukan dan mengarahkan permintaan. Contoh: Envoy, NGINX, Istio.
- **Decoupling of Clients and Services:**  
Pemutusan Keterkaitan: Klien tidak perlu mengetahui lokasi layanan yang tepat. Mereka hanya perlu mengirim permintaan ke load balancer atau service proxy, yang akan mengarahkan permintaan ke instance layanan yang sesuai.  
Sederhana: Ini menyederhanakan logika klien dan memungkinkan perubahan di sisi layanan tanpa mempengaruhi klien.
- **Centralized Management:**  
Manajemen Pusat: Service registry dan load balancer memungkinkan pengelolaan pusat lokasi layanan, yang membuat pemantauan dan manajemen layanan menjadi lebih mudah.  
Konfigurasi dan Kebijakan: Kebijakan routing dan load balancing dapat diterapkan di satu tempat dan berlaku untuk semua klien.
- **Health Checks and Monitoring:**  
Health Checks: Service registry sering melakukan pemeriksaan kesehatan secara berkala terhadap layanan yang terdaftar untuk memastikan mereka dapat diakses.  
Monitoring: Service registry dan load balancer dapat dilengkapi dengan kemampuan monitoring untuk memantau performa dan ketersediaan layanan.

Komunikasi Antar Microservices adalah aspek penting dalam arsitektur microservices, karena setiap microservice adalah unit independen yang sering kali perlu berinteraksi dengan microservice lain untuk menyelesaikan tugas atau memberikan layanan.



### Server Side Discovery

- Membuat server khusus sebagai router atau load balancer ke service
- Client hanya butuh terkoneksi ke router atau load balancer
- Jika jumlah node service bertambah atau berkurang, router yang hanya perlu dirubah, client tidak perlu berubah

Contoh Router atau Load Balancer ; Nginx, Apache HTTPD dan Traefik

### Kekurangan Server Side Discovery

- Tiap service harus memiliki router atau load balancer
- Agar tidak terjadi single point of failure, maka router atau load balancer harus di setup sebanyak 2 instance
- Cost biaya akan lebih mahal, karena 1 service harus menjalankan 2 router

