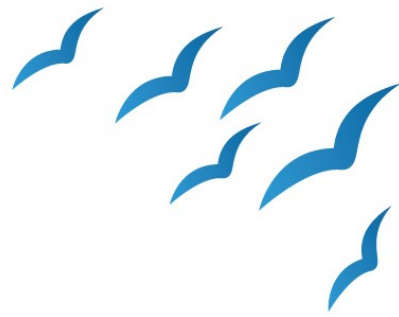


Pemrograman Framework Java

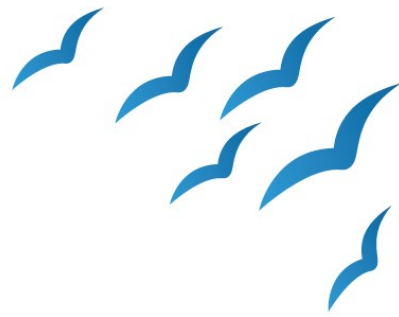
Web Service Restful





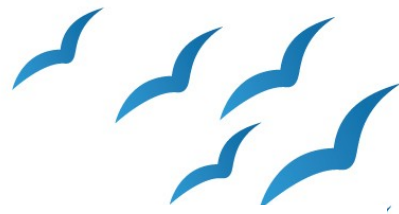
Sistem Terdistribusi

- Komponen-komponen perangkat keras dan perangkat lunak yang terpisah satu sama lain namun terhubung melalui sebuah jaringan
- Sistem ini dibuat untuk melakukan komputasi yang lebih besar dan kompleks

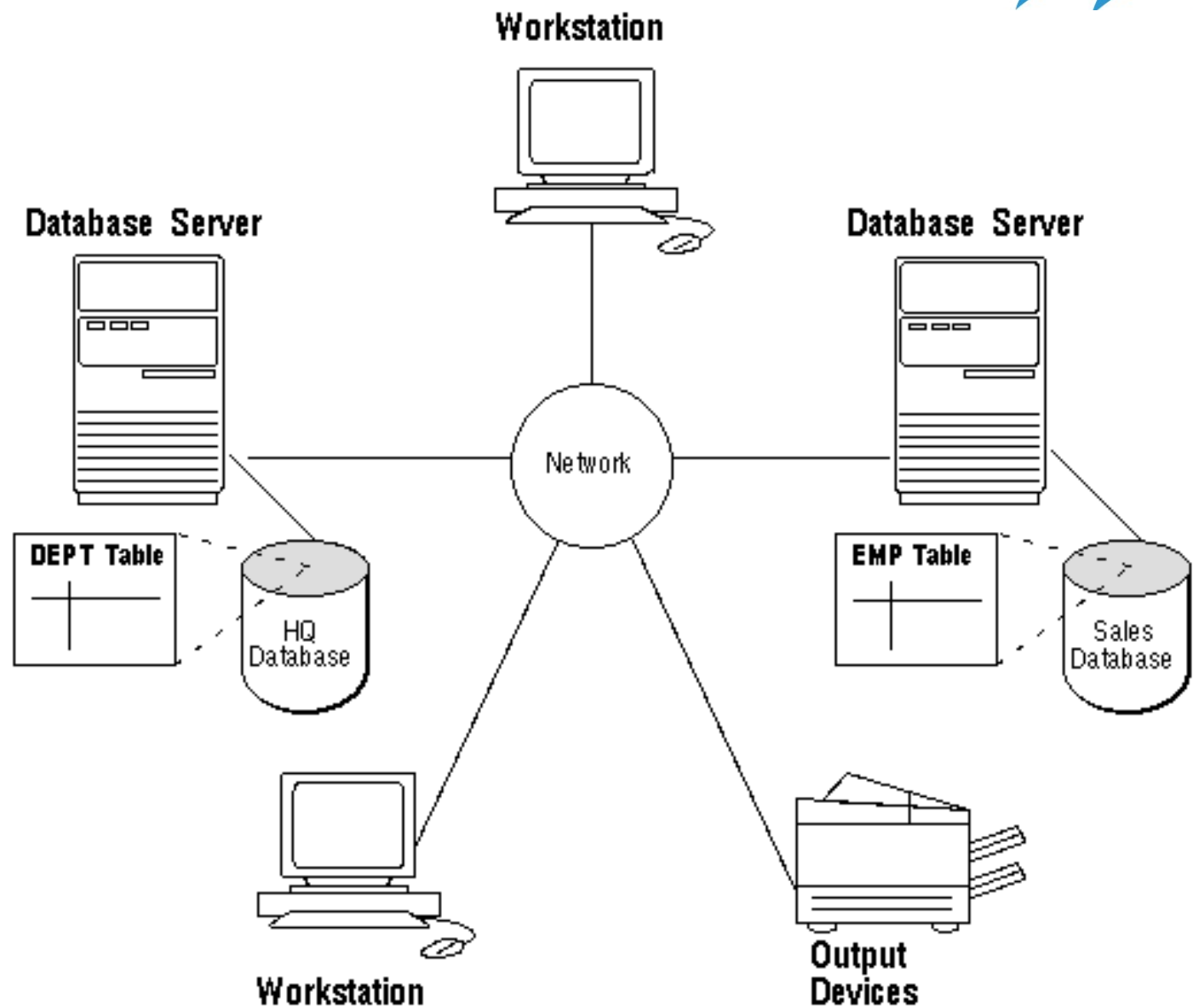


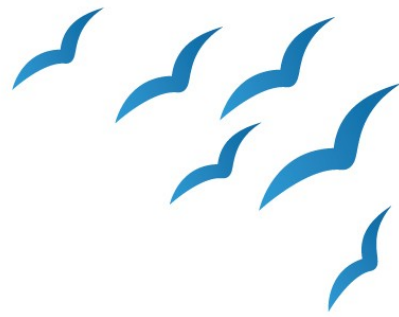
Lanjutan

- Inti dari sistem terdistribusi adalah kemajemukan perangkat keras dan lunak yang dapat berkomunikasi satu sama lain
- Dan konsep inilah yang digunakan oleh Web Service untuk berhubungan dengan berbagai macam bahasa pemrograman, salah satunya adalah Java



Ilustrasi



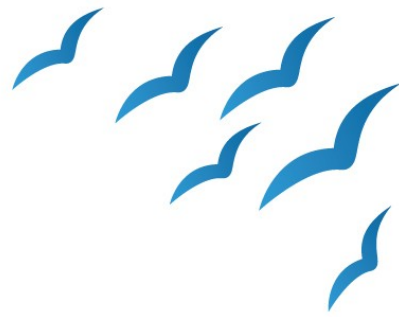


Mengenai Web Service

- Web Service memainkan peran yang sangat penting sebagai salah satu metode komunikasi antar aplikasii di dalam sebuah sistem terdistribusi
- Mirip dengan sistem terdistribusi, web service juga memiliki kumpulan protokol terbuka dan baku yang digunakan untuk pertukaran data

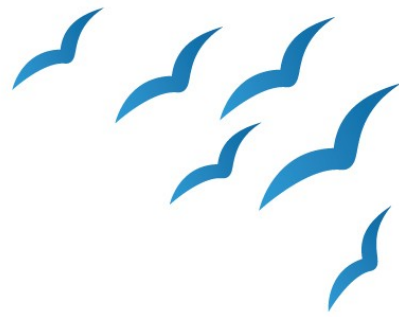
Ilustrasi





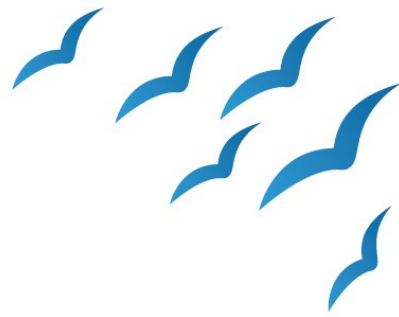
Lanjutan

- Web Service mengambil alih peran dari DCOM dan CORBA yang tidak kompatibel dengan platform internet
- Terdapat 2 pendekatan Web Service yang dapat dilakukan, yaitu Simple Object Access Protocol (SOAP) dan Representation State Transfer (REST)



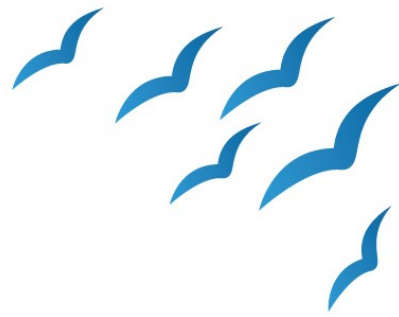
SOAP

- SOAP didefinisikan sebagai sebuah spesifikasi dari protokol komunikasi untuk pertukaran data berbasis XML.
- SOAP dapat menggunakan berbagai protokol misalnya HTTP, dan SMTP



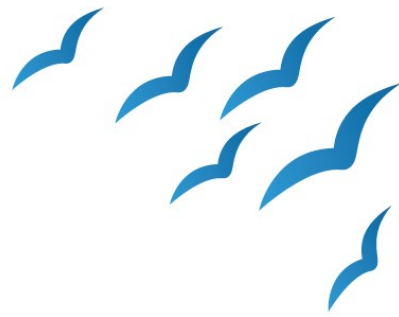
REST

- REST dikembangkan dengan pendekatan yang lebih baru mengenai Web Service dikarenakan SOAP tidak efisien dan sulit digunakan
- REST menjadi alternatif yang disukai programmer, dikarenakan REST hanya menggunakan URL yang sederhana



Batasan REST

- Client-Server : antarmuka memisahkan klien dari server dengan tujuan portabilitas tinggi
- Stateless : Tidak ada client context yang disimpan pada server di antara request-request yang dikirim
- Cacheable : penyimpanan data sementara dari response



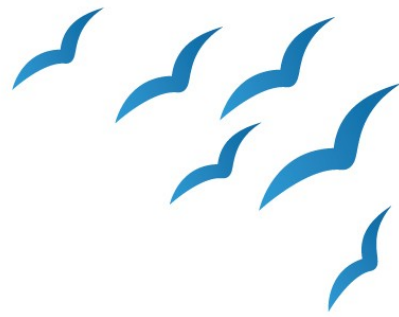
Lanjutan

- Layered System : Jika klien terhubung melalui perantara maka sistem itu harus mengaktifkan fitur pembagian program
- Code on Demand : Sesekali server dapat mengkustom klien dengan mengirim semacam program untuk dijalankan
- Uniform Interface : Keseragaman antar muka menyederhanakan dan menghilangkan ketergantungan

Menggunakan HTTP secara eksplisit

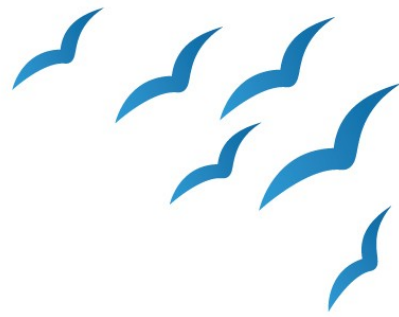


- REST meminta pengembang untuk menggunakan metode HTTP secara eksplisit dikarenakan konsistensi dengan definisi protokol.
- Hal ini menjadi desain dasar REST membangun satu-ke-satu mapping di antara operasi create, read, update, dan delete (CRUD)



Mapping REST

- Untuk membuat resource, gunakan POST.
- Untuk mengambil resource, gunakan GET.
- Untuk mengubah status atau data resource, gunakan PUT.
- Untuk membuang resource, gunakan DELETE.

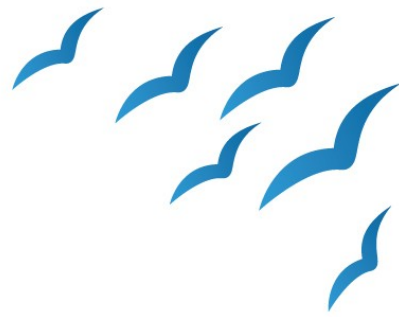


Ilustrasi



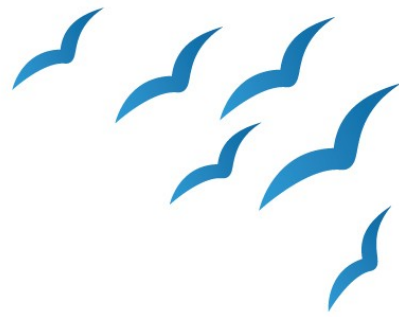
- GET
- POST
- PUT
- DELETE





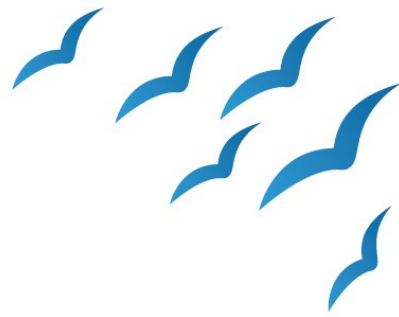
Kapan Menggunakan

- Dikarenakan pesan SOAP lebih berat dalam kontent dan memerlukan bandwidth yang besar, maka REST sebaiknya digunakan ketika bandwidth jaringan menjadi batasan
- Caching – Jika ada kebutuhan caching banyak request maka REST menjadi solusi yang sempurna



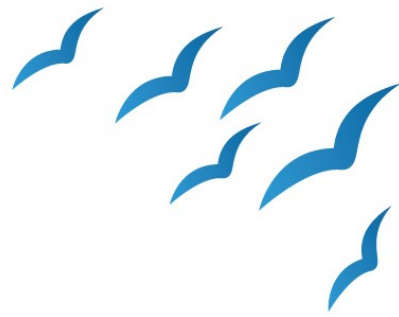
SOAP vs. REST API challenges

- WSDL file – Salah satu tantangan dari SOAP API adalah dokumen WSDL itu sendiri.
- Dokumen WSDL adalah sesuatu yang memberitahu klien semua operasi yang bisa dilakukan oleh web service.



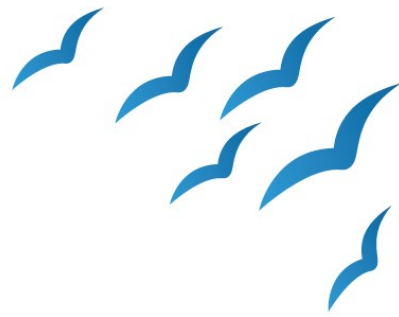
Lanjutan

- Tantangan dari REST adalah
- Kurangnya Keamanan – REST tidak menyediakan keamanan layaknya SOAP, ketika mengirimkan data penting, REST adalah mekanisme terburuk dalam Web Service.



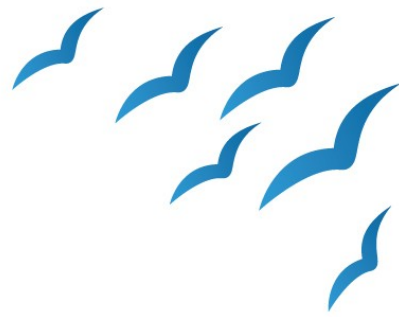
Lanjutan

- Kurangnya state – Sebagian besar dari aplikasi web memerlukan mekanisme stateful. Sebagai contoh, ketika anda membeli sebuah situs yang memiliki mekanisme shopping cart, maka dia akan memerlukan berapa jumlah dalam shopping cart sebelum pembelian dilakukan.



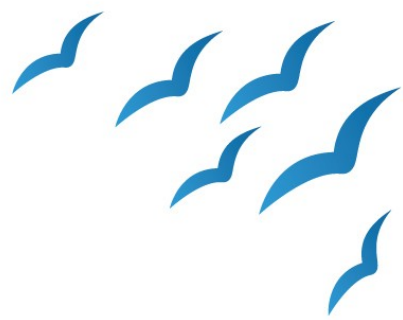
Pembuatan REST Service

- Spring Framework mendukung dua cara pembuatan RESTful service:
 - Menggunakan MVC dengan ModelAndView
 - Menggunakan konversi pesan HTTP

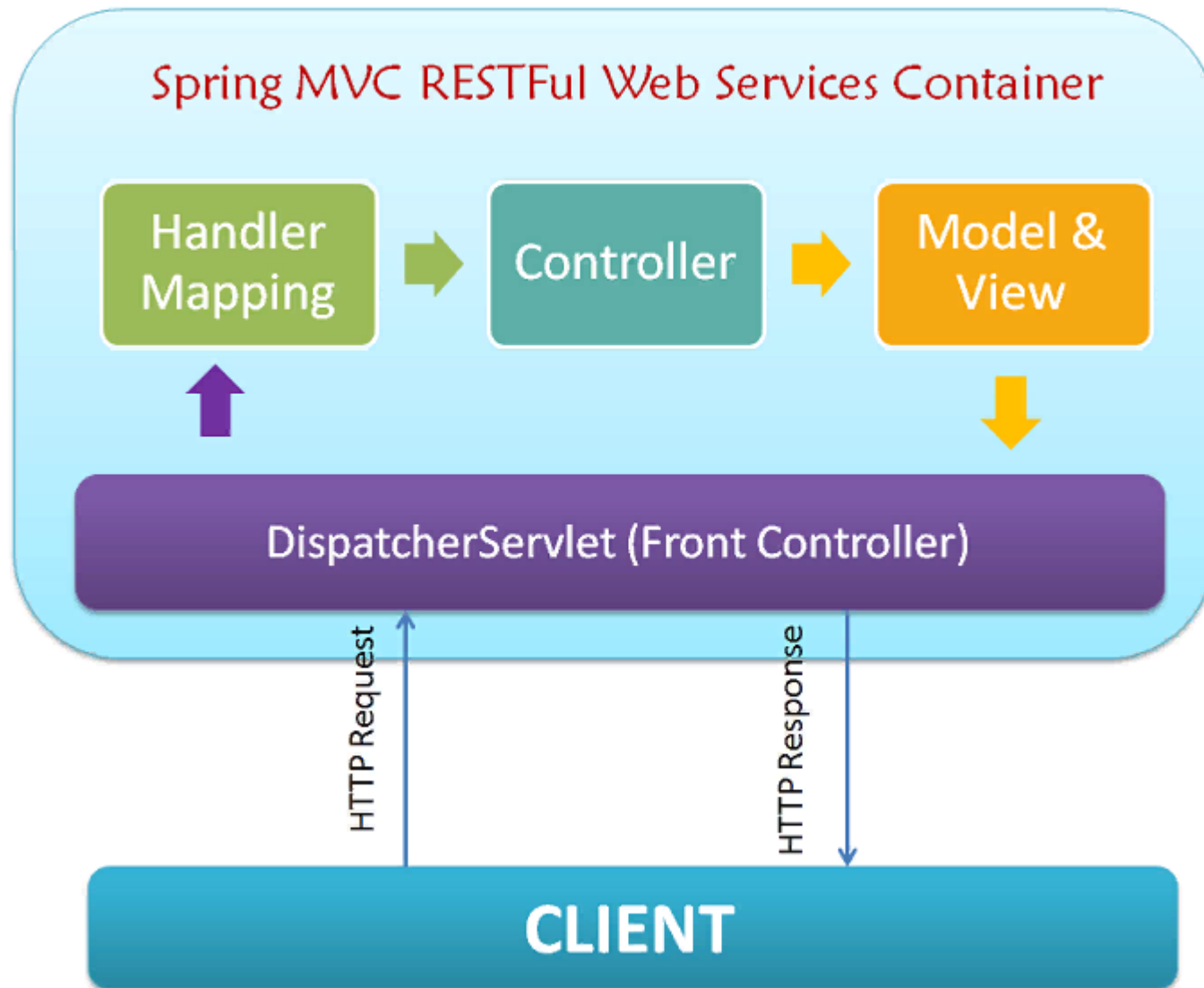


Lanjutan

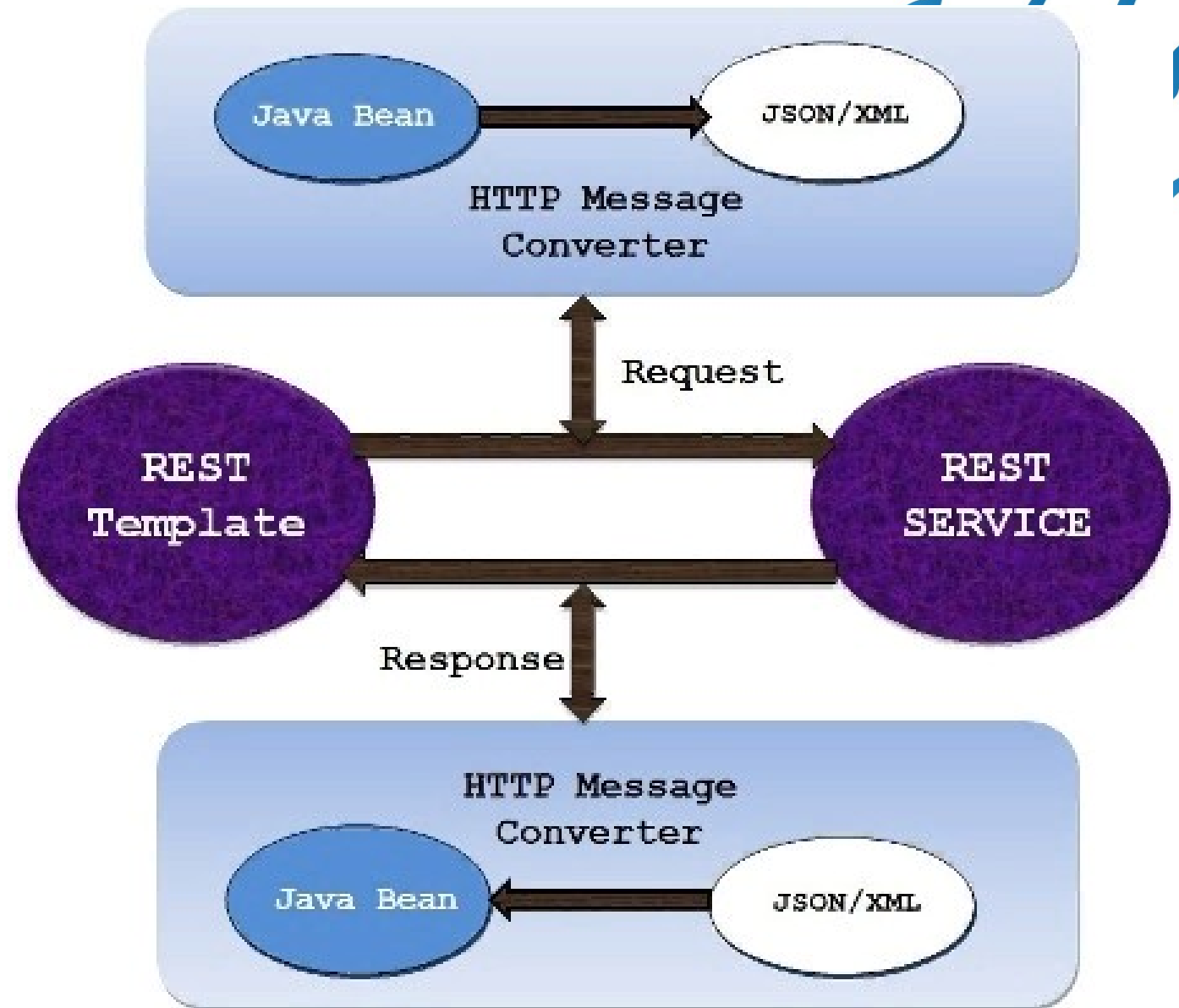
- Pendekatan Model-And-View yang lebih tua dan terdokumentasi lebih baik, tetapi juga lebih memerlukan konfigurasi yang berat
- Pendekatan baru yang berdasarkan dari HTTPMessageConverter dan anotasi ini lebih ringan, mudah diimplementasikan, dan konfigurasi minimal

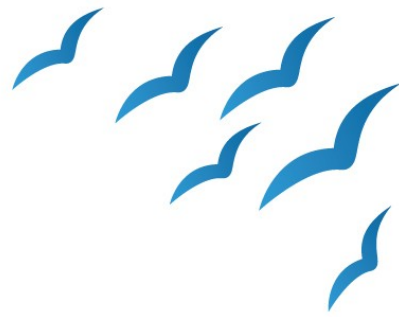


Ilustrasi



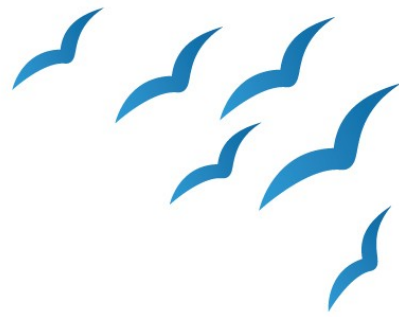
Ilustrasi





Keuntungan REST

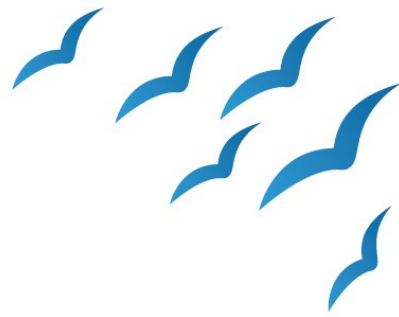
- Sempel untuk dibangun dan adaptasi
- Penggunaan resource yang rendah
- Instance Proses dibuat secara eksplisit
- Dengan initial URI, klien tidak memerlukan rute informasi
- Klien bisa memiliki generic 'listener' interface untuk notifikasi



Lanjutan

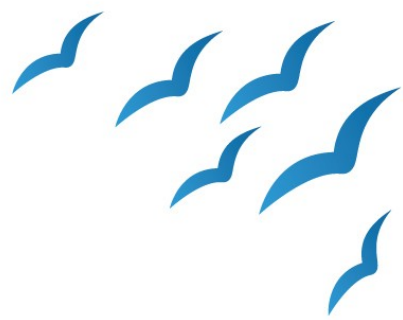
Dengan membangun di atas HTTP, REST APIs menyediakan kemudahan membangun fleksibel APIs yang bisa:

- Support backward compatibility
- Evolvable APIs
- Scaleable services
- Securable services
- A spectrum of stateless to stateful services



REST Pada Spring

- Spring memiliki fitur untuk menggunakan REST Api dalam pengembangan aplikasi web
- Dasar dari REST API yang dapat digunakan di dalam Spring seperti membuat aplikasi web, REST pagination, dan Error Handling



Ilustrasi

