

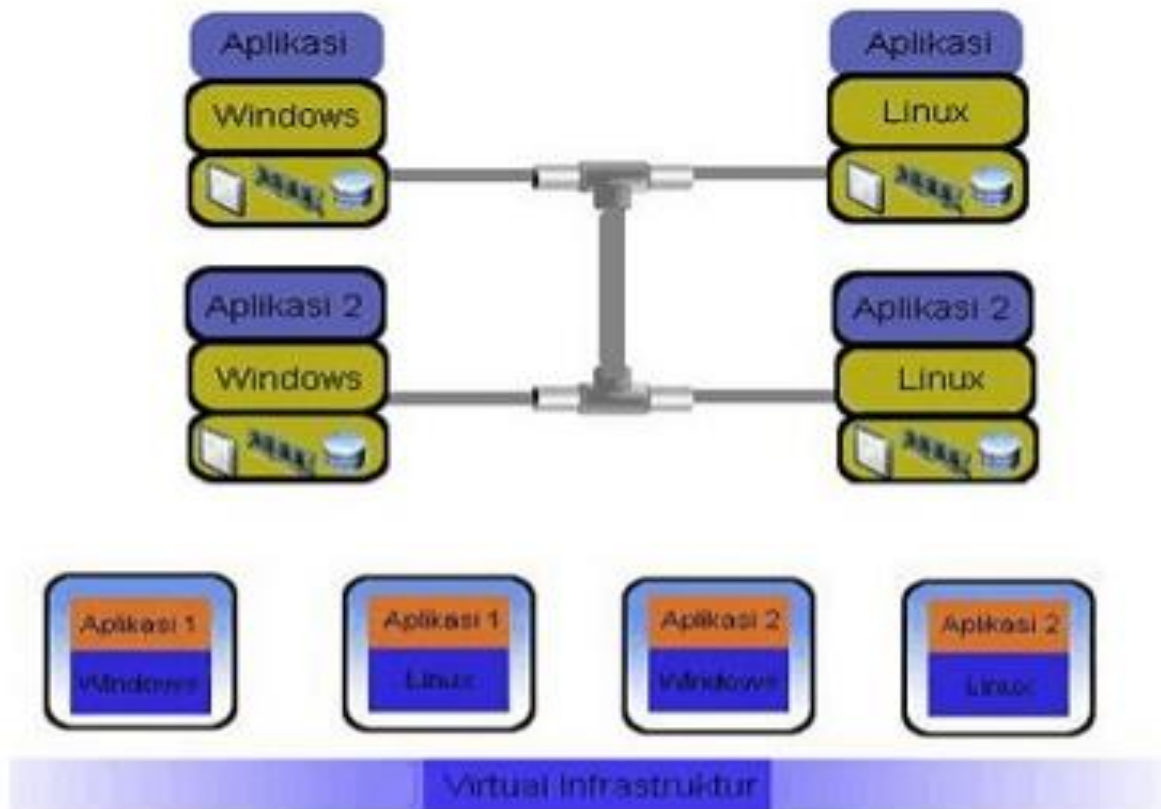
Pertemuan 9

Utility dan Web Service

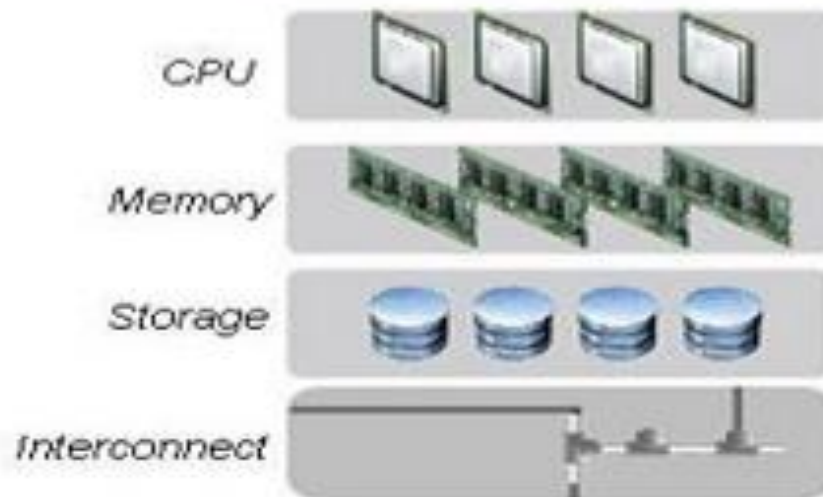
I. Utility Computing

Cloud computing tidak hanya melibatkan sisi aplikasi atau perangkat lunak saja, tetapi juga melibatkan perangkat keras atau hardware dan sumber daya penunjang. Seperti yang telah kita ketahui layanan SaaS lebih berfokus pada aplikasi atau perangkat lunak, sedangkan pada infrastruktur sebagai layanan utility computing. Layanan utility computing dikemas oleh provider dalam bentuk teknologi virtualisasi dan dikenal sebagai layanan IaaS (Infrastructure as a Service).

Utility Computing (Lanjutan)



Utility Computing (Lanjutan)



Gambar 1. Infrastructure Virtual
Sumber : (herwin:2011)

Utility Computing (**Lanjutan**)

Masing–masing sistem operasi (windows dan linux) menggunakan sumber daya komputer yang sama. Sistem operasi pada gambar tersebut bukanlah sesuatu yang special sebagai peranan utama dalam infrastruktur virtualisasi. Sistem operasi hanya sebagai perantara untuk dapat menjalankan virtual mesin. Peranan utama dalam infrastruktur virtualisasi adalah hypervisor. Hypervisor merupakan software yang menggantikan fungsi utama dari operating sistem ketika operating sistem selesai menjalankan virtual mesin. Hypervisor diasumsikan sebagai virtual machine manager, yang didesign untuk dapat menjalankan virtual mesin lainnya dan menjalankan sistem operasi dari awal seperti ketika komputer dinyalakan.

Utility Computing (lanjutan)

Dengan teknologi virtualisasi, pengguna atau penyewa IaaS dapat mengakses dan menggunakan seluruh sumber daya komputer dan seluruh sumber daya lainnya yang tersedia di dalam cloud sesuai kebutuhan dan keinginan pengguna. Teknologi virtualisasi memungkinkan untuk diimplementasikan berbagai aplikasi dengan tujuan yang beragam dalam 1 platform atau aplikasi, seperti storage computing, image manipulation, parallel processing, content distribution, aplikasi web dan sebagainya.

Utility Computing (lanjutan)

Dalam menawarkan layanan IaaS kepada pengguna atau penyewa, provider membagi IaaS dalam beberapa kategori layanan yaitu:

1. Layanan penyimpanan dan komputasi virtual: yaitu VMware rental, penyimpanan online (Online Storage).
2. Layanan kustomise: yaitu server template.
3. Layanan automasi dan control: yaitu automation.
4. Layanan penghubung: yaitu remote control, web 2.0.
5. Layanan monitoring: yaitu monitor secara fisik objek yang diinginkan (posisi koordinat bumi, peta, kamera).

Utility Computing (lanjutan)

6. Layanan optimasi objek: yaitu virtualisasi network, virtualisasi penyimpanan, virtualisasi server.
7. Layanan pengukuran objek: yaitu pengukuran fisik suatu objek.
8. Layanan integrated dan kombinasi objek yaitu load balance.
9. Layanan security: yaitu enkripsi data penyimpanan, VM isolation, VLAN dan SSL/SSH.

Utility Computing (lanjutan)

Secara infrastruktur, penerapan teknologi virtualisasi pada IaaS di cloud computing memberikan beberapa kemudahan & keuntungan bagi penyewa.

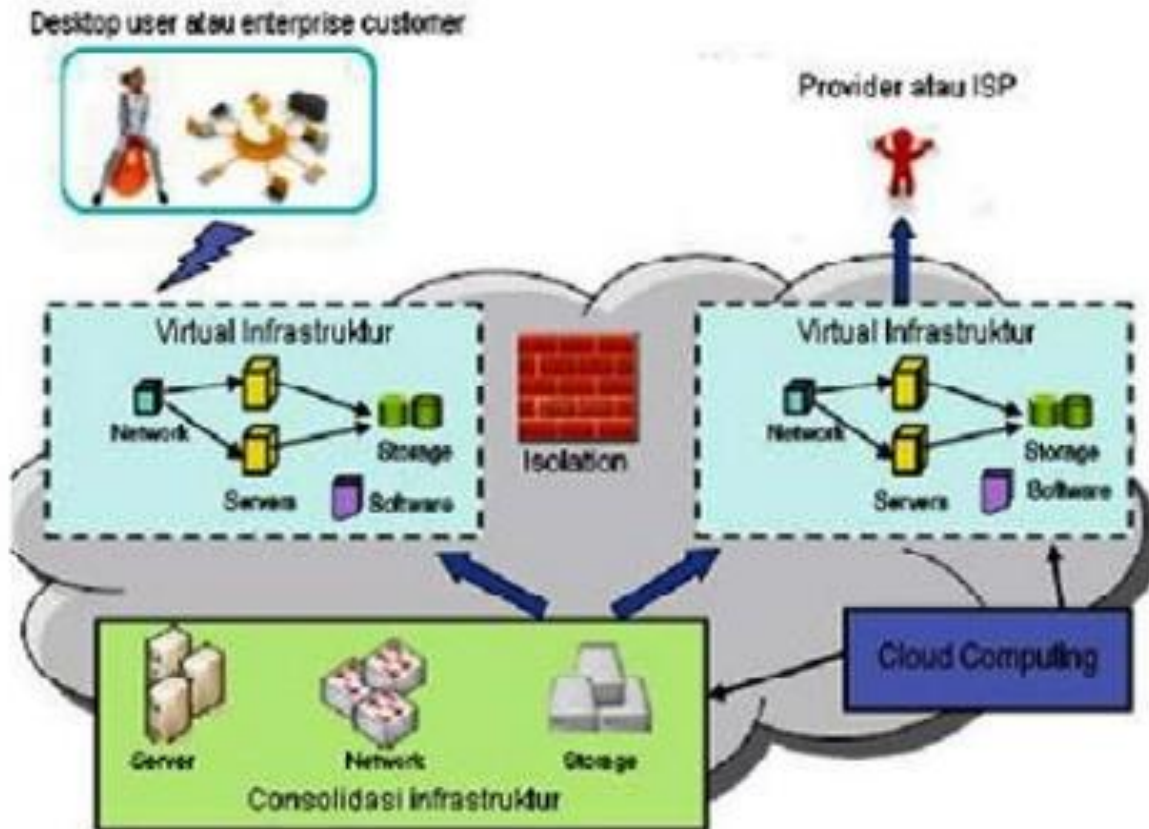
| | Co-location | Server Sewa | IaaS pada cloud |
|-----------------------------------|---|---|--|
| Performance | Sangat tergantung pada spesifikasi hardware | Sangat tergantung pada spesifikasi hardware | Performance yang terjamin |
| Biaya yang harus dikeluarkan | Investasi pada server, bandwidth dan rak penyimpanan server | Bandwidth dan biaya penyewaan server | Biaya CPU, memory, bandwidth dan media penyimpanan berdasarkan pemakaian |
| Ketersediaan | Sangat tergantung pada spesifikasi hardware | Sangat tergantung pada spesifikasi hardware | Ketersediaan yang tinggi |
| Pengembangan dan perluasan sistem | Dilakukan secara manual | Dilakukan secara manual | Pengembangan dan perluasan sistem dilakukan secara otomatis |
| Manajemen sistem | Instalasi atau set up hardware, konfigurasi | Instalasi atau set up hardware dan | Instalasi sistem operasi dan software |

Utility Computing (lanjutan)

| | | | |
|------------------------------------|---|---|--|
| | sistem operasi cukup rumit dan dilakukan secara manual | konfigurasi sistem operasi cukup rumit dan dilakukan secara manual | dilakukan secara otomatis. Monitor dan kontrol jarak jauh. |
| Staff atau karyawan | Membutuhkan keahlian khusus dan specific | Membutuhkan keahlian khusus dan specific | Tidak membutuhkan keahlian khusus |
| Sisi maintenance dan pengoperasian | Maintenance dilakukan di tempat. Pengoperasian terkadang perlu dilakukan di tempat. | Maintenance dilakukan di tempat. Pengoperasian terkadang perlu dilakukan di tempat. | Dapat dilakukan melalui aplikasi web dan instan. |

Jantung dari teknologi cloud computing adalah virtualisasi, dimana virtualisasi dapat diterapkan pada 2 sisi yaitu pada sisi provider dan sisi pengguna seperti pada gambar

Utility Computing (lanjutan)



Gambar 2. Desktop Pengguna

Sumber : (herwin:2011)

Utility Computing (lanjutan)

Beberapa software virtulisasi seperti VMware, citrix dan sebagainya mempunyai kemampuan untuk menciptakan fungsi lain yang disebut sebagai virtual desktop interface (VDI). Virtual desktop interface (VDI) menciptakan session untuk client atau user di dalam server, dan mengirasehingga user dapat berinteraksi dengan server seakan client atau user tersebut berada di dalam server itu sendiri. Perbedaan yang cukup signifikan antara software remote dengan virtual PC imkan virtual PC tersebut kepada client atau user

Utility Computing (lanjutan)

Software remote adalah software yang dapat digunakan untuk melakukan pengendalian jarak jauh ke satu komputer atau satu server dalam satu koneksi hanya untuk satu user atau client. Jika satu komputer atau satu server diakses oleh lebih dari dua user maka komputer atau server yang diakses secara remote akan memutuskan salah satu koneksi dari dua koneksi yang terjadi.

Utility Computing (lanjutan)

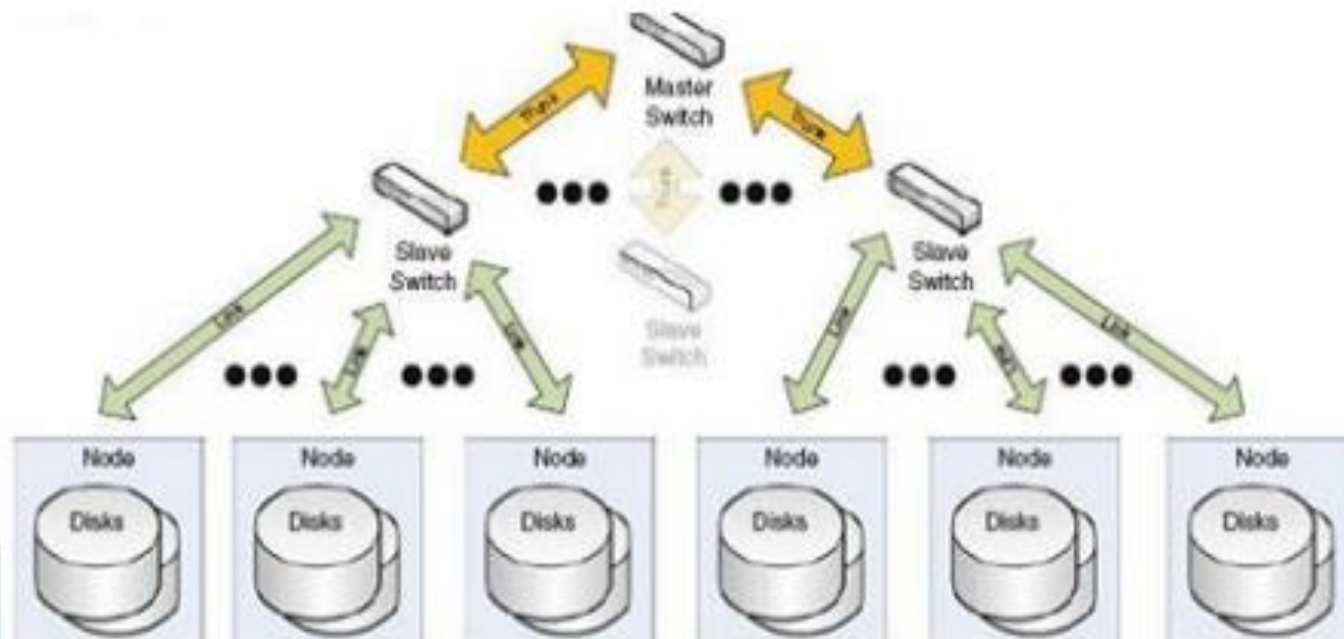
Software remote hanya software atau aplikasi penghubung ke komputer lain dan tidak dapat berfungsi untuk menciptakan komputer di dalam komputer itu sendiri. user terkoneksi dan menggunakan layanan IaaS ke server provider melalui virtual desktop interface (VDI) di internet. Sedangkan pada sisi provider, provider melakukan konfigurasi server melalui jalur yang sama (VDI) di internet.

Untuk dapat menerapkan teknologi virtualisasi di cloud maka server yang sudah diimplementasikan teknologi virtualisasi diletakkan di dalam cloud (private cloud atau public cloud) sebagai back end infrastruktur. Dari perspektif ini, sumber daya teknologi virtualisasi atau virtual resources di dalam cloud diasumsikan sebagai sumber daya komputer yang bersifat independent atau mandiri termasuk lokasi dari sumber daya itu sendiri.

Utility Computing (lanjutan)

Infrastruktur juga memegang peranan utama untuk memastikan semua komponen bekerja dengan baik dalam kondisi multi tenant dan bertanggung jawab terhadap segala aktifitas yang terjadi. Seperti yang sudah dijelaskan sebelumnya bahwa teknologi virtualisasi merupakan jantung utama dari cloud computing, dimana teknologi virtualisasi hanyalah berupa aplikasi atau software. Teknologi virtualisasi tidak dapat berjalan sempurna tanpa didukung dengan infrastruktur yang baik dan solid. Teknologi virtualisasi memungkinkan untuk diterapkan redundancy, replication atau cluster, dan workload balancing.

Utility Computing (lanjutan)



Gambar 3. Ilustrasi infrastruktur yang baik dan solid

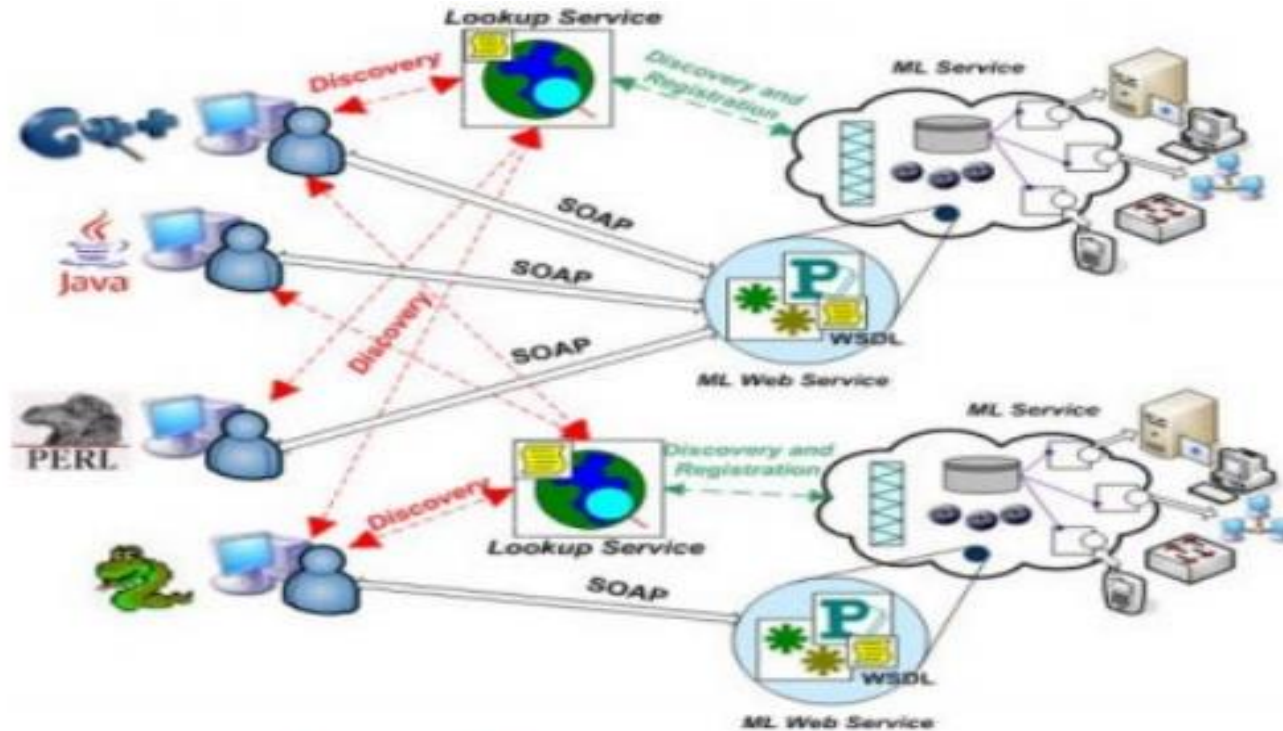
Sumber : (herwin:2011)

II. Web Service

Kemampuan unik dari web service adalah membantu para programmer untuk membuat suatu aplikasi berbasis web dengan fungsi lain di atas platform web itu sendiri.

Dalam beberapa kasus, coding – coding yang dihasilkan oleh programmer yang menyewa layanan ini membagikan (share) dan dikumpulkan dalam penyimpanan data yang dikelola oleh provider

Web Service (lanjutan)



Gambar 4. Ilustrasi Layanan web service

Sumber : <http://id.wikipedia.org/wiki>

Web Service (lanjutan)

Web Service merupakan fenomena yang sangat panas saat ini karena, banyak kelebihan yang ditawarkan oleh Web Service terutama interoperabilitas tinggi dan penggunaannya yang dapat diakses kapanpun dan dimanapun selama mesin kita terhubung oleh jaringan internet salah satunya.

Web Service (lanjutan)

Pengertian Web Service Sepenuhnya berdasarkan standard web dan xml. Web Service dapat membantu : Perantara pada integrasi platform sepanjang eksekusi mesin virtual. Integrasi antara Web dan OO middleware. Integrasi dari aliran kerja terisolasi dan service-service (Web Services Flow Language - WSFL). Pertukaran data pada aplikasi yang berbeda-beda (X- Schema, XSLT ++) (Masa depan: standarisasi dari info konteks antara web servis dan klien – integrasi servis horizontal).

Web Service (lanjutan)

Pemain utama dan standard-standard :

Microsoft: .NET SUN: Open Net Environment (ONE) IBM: Web Service Conceptual Architecture (WSCA) W3C: Web Service Workshop Oracle: Web Service Broker Hewlett-Packard: Web Service Platform Kemampuan aplikasi, fungsi atau operasi yang di ekspos untuk program lain melalui standard yang terbuka, dan interoperable. “payloads” didefinisikan sebagai XML. “transports” melalui http atau Internet protocol terbuka lainnya. Data diakses dari berbagai bahasa pemrograman , platform hardware atau system operasi. Middleware dari Internet.

Web Service (lanjutan)

Keuntungan penggunaan Web Service

- ☐ Format penggunaan terbuka untuk semua platform.
- ☐ Mudah di mengerti dan mudah men-debug.
- ☐ Dukungan interface yang stabil.
- ☐ Menggunakan standard-standard “membuka service sekali” dan mempunyai pemakai banyak.
- ☐ Mudah untuk menengahi pesan-pesan proses dan menambahkan nilai.

Web Service (lanjutan)

Routing and pengiriman.

- ☐ Security.
- ☐ management and monitoring.
- ☐ schema and service design.
- ☐ Akselerasi.
- ☐ mudah untuk mengembangkan dengan semantic transport tambahan.
- ☐ Terbuka, standard-standard berbasis teks.
- ☐ Pencapaian modular.
- ☐ Tidak mahal untuk diimplementasikan (relatif).
- ☐ Mengurangi biaya integrasi aplikasi enterprise.
- ☐ Implementasi yang incremental.

Web Service (lanjutan)

A. Mekanisme Kerja

Sebuah layanan Web adalah metode komunikasi antara dua perangkat elektronik melalui jaringan.



Gambar 5. Web Arsitektur

Sumber : <http://id.wikipedia.org/wiki>

Web Service (lanjutan)

The W3C mendefinisikan "layanan Web" sebagai "sebuah sistem perangkat lunak yang dirancang untuk mendukung interoperable mesin-ke-mesin interaksi melalui jaringan Bahasa. ini memiliki antarmuka yang dijelaskan dalam mesin-processable format (khusus Web Services Deskripsi WSDL). Sistem lain berinteraksi dengan layanan Web dalam cara yang ditentukan oleh deskripsi dengan menggunakan SOAP pesan, biasanya disampaikan menggunakan HTTP dengan serialisasi XML dalam hubungannya dengan standar Web-terkait lainnya. "

Web Service (lanjutan)

W3C juga menyatakan, "Kita dapat mengidentifikasi dua kelompok utama layanan Web, REST layanan Web-compliant, di mana tujuan utama pelayanan ini adalah untuk memanipulasi representasi sumber daya XML Web menggunakan seragam set "stateless" operasi; dan sewenang-wenang layanan Web, di mana layanan akan mengekspos serangkaian operasi sewenang-wenang. "

Web Service (lanjutan)

B. Big layanan Web

"Big layanan Web" menggunakan Extensible Markup Language (XML) pesan yang mengikuti SOAP standard dan telah populer dengan usaha tradisional. Dalam sistem seperti itu, sering kali ada yang bisa dibaca deskripsi mesin operasi yang ditawarkan oleh layanan ditulis dalam Web Services Description Language (WSDL). Yang terakhir adalah tidak merupakan persyaratan dari sebuah titik akhir SOAP, tetapi merupakan prasyarat untuk otomatis sisi klien. SOAP kerangka (kerangka kerja seperti Apache axis2 , Apache CXF , dan Spring menjadi pengecualian). Beberapa organisasi industri, seperti WS-I , mandat baik SOAP dan WSDL dalam definisi mereka tentang layanan Web.

Web Service (lanjutan)

C. Web API



Gambar 6. Layanan web dalam service-oriented architecture

Sumber : <http://id.wikipedia.org/wiki>

Web Service (lanjutan)

C. Web API

Web API adalah pembangunan di layanan Web (dalam gerakan yang disebut Web 2.0) dimana penekanan telah bergerak menjauh dari SOAP layanan berbasis terhadap Transfer (REST) komunikasi berbasis. REST tidak memerlukan layanan XML, SOAP, atau WSDL layanan-API definisi. Web API memungkinkan kombinasi dari berbagai layanan web ke dalam aplikasi baru yang dikenal sebagai mashup.

Web Service (lanjutan)

C. Web API

Web API adalah pembangunan di layanan Web (dalam gerakan yang disebut Web 2.0) dimana penekanan telah bergerak menjauh dari SOAP layanan berbasis terhadap Transfer (REST) komunikasi berbasis. REST tidak memerlukan layanan XML, SOAP, atau WSDL layanan-API definisi. Web API memungkinkan kombinasi dari berbagai layanan web ke dalam aplikasi baru yang dikenal sebagai mashup.

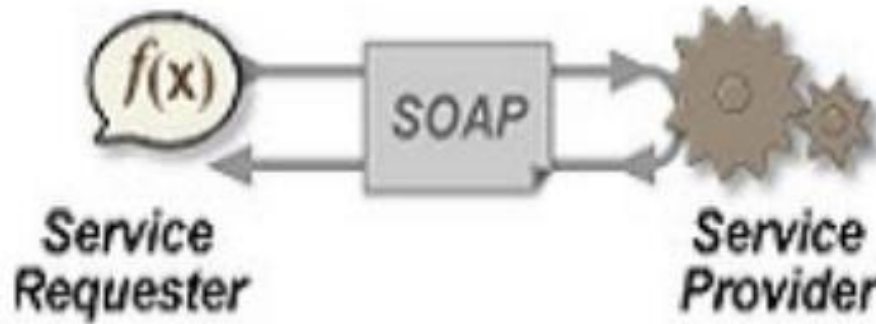
Web API (lanjutan)

Ketika digunakan dalam konteks pengembangan Web , Web API ini biasanya kelompok yang telah ditetapkan dari Hypertext Transfer Protocol (HTTP) pesan permintaan bersama dengan definisi struktur pesan respon, biasanya dinyatakan dalam sebuah Extensible Markup Language (XML) atau JavaScript Object Notation (JSON) format. Saat menjalankan layanan Web komposit, setiap layanan sub dapat dianggap otonom. User tidak memiliki kendali atas layanan ini. Juga Web services sendiri tidak dapat diandalkan, penyedia layanan dapat menghapus, mengubah atau memperbarui jasa mereka tanpa memberikan pemberitahuan kepada pengguna. Toleransi kehandalan dan kesalahan tidak didukung; kesalahan mungkin terjadi selama eksekusi.

Web API (lanjutan)

penanganan Eksepsi dalam konteks layanan Web adalah masih merupakan masalah dalam penelitian terbuka. Tetap saja dapat ditangani dengan menanggapi dengan obyek kesalahan ke klien.

D. Prosedur Remote Panggilan



Gambar 7. Arsitektur elemen yang terlibat dalam XML-RPC.

Sumber : <http://id.wikipedia.org/wiki>

RPC Web layanan menyajikan fungsi didistribusikan (atau metode) antarmuka panggilan yang akrab bagi banyak pengembang. Biasanya, unit dasar layanan RPC Web adalah operasi WSDL.

Prosedur Remote Panggilan (lanjutan)

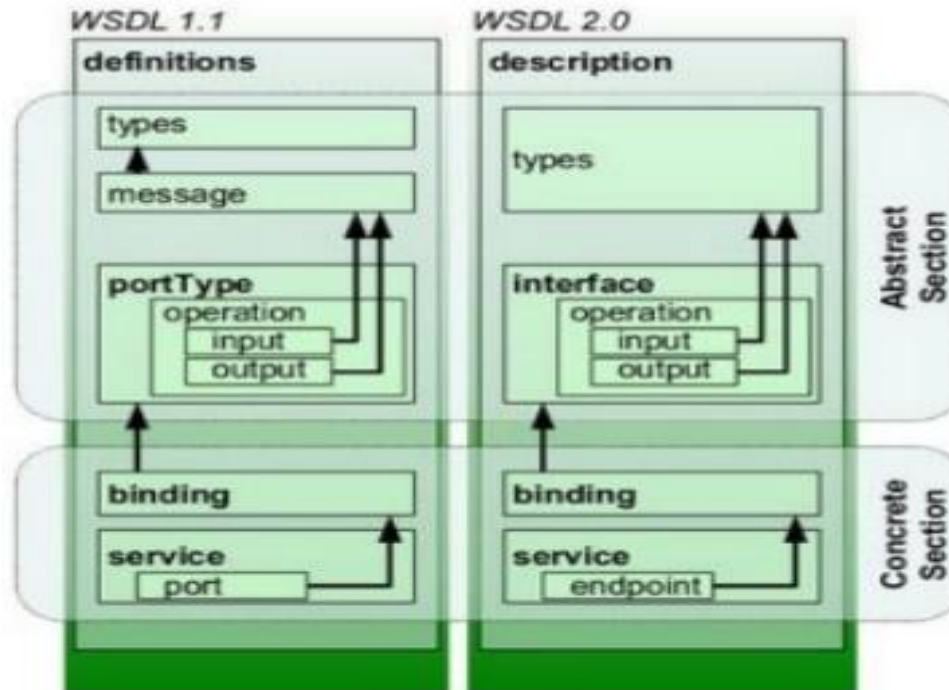
Web pertama pelayanan alat difokuskan pada RPC, dan sebagai hasilnya gaya ini secara luas digunakan dan didukung. Namun, kadang-kadang dikritik karena tidak longgar ditambah, karena sering dilaksanakan oleh jasa pemetaan langsung ke-spesifik fungsi bahasa atau panggilan metode. Banyak vendor merasa pendekatan ini menjadi buntu, dan mendorong untuk RPC untuk menjadi batasan dalam WS-I Basic Profile. Pendekatan-pendekatan lain dengan hampir fungsi yang sama seperti RPC adalah Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA), Microsoft's Distributed Component Object Model (DCOM) atau Sun Microsystems's Java/Remote Method Invocation (RMI).

E. Arsitektur Berorientasi Layanan

Layanan Web juga dapat digunakan untuk mengimplementasikan arsitektur sesuai dengan konsep service-oriented architecture (SOA), di mana unit dasar komunikasi adalah pesan, bukan operasi. Hal ini sering disebut sebagai "pesan-berorientasi" layanan. SOA Web layanan yang didukung oleh sebagian besar vendor software utama dan analis industri. Tidak seperti layanan Web RPC, kopling longgar lebih mungkin, karena fokusnya adalah pada "kontrak" yang WSDL menyediakan, bukan detail implementasi yang mendasarinya

Arsitektur Berorientasi Layanan (lanjutan)

Middleware analis menggunakan bus pelayanan perusahaan yang menggabungkan pesan-berorientasi pengolahan dan layanan Web untuk membuat event-driven SOA . Salah satu contoh sumber terbuka ESB adalah Mule, satu lagi adalah Open ESB



Gambar 8. Representasi konsep didefinisikan oleh WSDL 1.1 dan WSDL 2.0 dokumen.

Sumber : <http://id.wikipedia.org/wiki>

F. Representasi Transfer (Rest)

REST berusaha untuk menggambarkan arsitektur yang menggunakan protokol HTTP atau serupa oleh menghambat antarmuka untuk satu set terkenal, operasi standar (seperti GET, POST, PUT, DELETE untuk HTTP). Di sini, fokusnya adalah pada berinteraksi dengan stateful sumber daya, daripada pesan atau operasi. Sebuah arsitektur yang berbasis pada REST dapat menggunakan WSDL untuk menggambarkan pesan SOAP melalui HTTP, dapat diimplementasikan sebagai sebuah abstraksi murni di atas SOAP (misalnya, WS-Transfer), atau dapat dibuat tanpa menggunakan SOAP di semua.

F. Representasi Transfer (lanjutan)

WSDL versi 2.0 menawarkan dukungan untuk mengikat kepada semua metode permintaan HTTP (tidak hanya GET dan POST seperti pada versi 1.1) sehingga memungkinkan implementasi yang lebih baik dari layanan Web tenang. Namun, dukungan untuk spesifikasi ini masih miskin dalam pengembangan perangkat lunak kit yang sering menawarkan alat hanya untuk WSDL 1.1