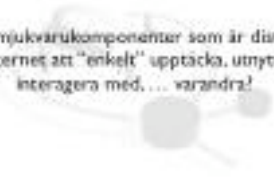


Tjänsteorienterade arkitekturer

Hur får vi mjukvarukomponenter som är distribuerade över internet att "enkelt" upptäcka, utnyttja, och interagera med, ... varandra?



Samverkande system

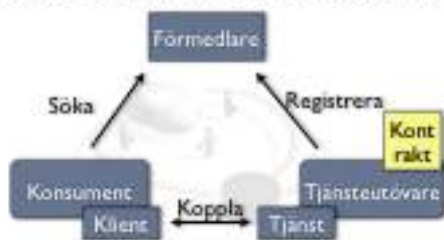
- Vi vill ha
 - Komponenter som erbjuder tjänsterlogisk funktion
 - Godtyckligt kommunikationsmönster
 - Hjärt/bort kopplade komponenter
 - Upptäcka/söka efter tjänster
 - Mechanismer för att administrera komponenterna



Bakgrund



Service Oriented Architecture



Bakgrund



Kärnfunktionalitet

- Transport – dataformat och protokoll för att kommunicera med "tjänsten"
- Beskriva – språket för att beskriva tjänsten för andra tjänster (maskinläsbart format)
- Upptäcka/leta – mekanismen för att registrera och åskådliggöra tjänster och för att hitta tjänster och dess beskrivningar



Kommunikationsmekanismer

- Socketprogrammering
 - Applikationsprotokoll
 - TCP/IP -stacken
- "Middleware" för distribuerade system
 - Java RMI
 - CORBA
 - DCE
 - DCOM



"Middleware"

- Alla har sina egna protokoll och format för att implementera SOA
- Fungerar bra inom plattformen, men...
 - Svårt att få tjänster i olika plattformar att prata
 - Kräver att en viss "middleware" är installerat på alla burlar



“Internet middleware”

- “Web-services” är en ny form av “middleware”
- Kräver ingen ny uppsättning protokoll
- XML är grunden för alla interna protokoll
- Vilken mjukvara som helst skriven i vilket programspråk på vilken plattform som helst kan tolka WS-beskrivningar, meddelanden och mekanismer för sökning/upptäcka
- Alla applikationer kan tolka SOAP-meddelanden genom att använda standard XML-verktyg



Vad är det?

“Web-services are loosely coupled software components delivered over standard Internet technologies”

“From a technical point of view, WS can be seen as an application API that can be invoked by an URL”

“ I think of a WS as an arbitrarily grouped set of resources intended for machines (not human) manipulation, as a “web site” is an arbitrarily grouped set of resources intended for human manipulation”



Gemensam vy...

- Man kommer åt dom via internet protokoll
- En mjukvarukomponent (digital artefakt)
- Ämnad för andra mjukvarukomponenter
- Löst kopplade
- URI
- URL
- WS är en arkitektur inte en samling med teknologier



Ytterligare en bild?

- Webben – ett gigantiskt informationsrum som ger access till web-resurser
- Tjänst – är en resurs som exponerar dess funktionalitet genom ett programmeringsgränssnitt (API)
- Web-service – en tjänst som kan identifieras med hjälp av en URI och accessas via web-protokoll i enlighet med vad som står i kontraktet som beskriver dess API



Komponera tjänster

- En tjänst representerar en unik applikation och funktionalitet, ...
- Varje sådan enhet kan användas och är synlig från varje nod på Internet
- Modulärt
 - Enheter kan kombineras och sättas samman till mer komplexa tjänster



WS-Tekniker

- XML-baserade protokoll
 - Transport
 - SOAP – Simple Object Access Protocol
 - Beskrivning
 - WSDL – Web Services Description Language
- Register/Upptäcka
 - UDDI



Transport (SOAP)

- Ett ramverk för att sända meddelanden mellan applikationer
- Data kan vara ett RPC anrop eller svar eller bara ett XML dokument
- Ett meddelande består av
 - **Header** (ej obligatorisk) – tillförlidighet, routing, betalningsmekanismer, säkerhet, ...
 - **Body** – innehåller själva data
- **SOAP Transport Binding Framework** - knyter till ett visst protokoll (HTTP, SMTP, IMAP, POP3, FTP, ...)
- **SOAP Serialization Framework** – hur XML-dokumentet ska tolkas mot applikationstyper m h a kodningscheman
- **SOAP Attachments** - möjlighet att skicka med icke XML-data med ett meddelande



Beskrivning (WSDL)

- Ett antal definitioner uttrycka i XML
- Ett WSDL dokument beskriver
 - **Vad** tjänsten har för funktionalitet
 - "Abstract interface" – är mängd med operationer och dess in/ut meddelanden (format och datatyper på varje element)
 - **Hur** tjänsten kommunicerar
 - "Concrete binding" – RPC/dokument, XML-schema, vilka "SOAP-headers", transportprotokoll, ...
 - **Var** man hittar tjänsten
 - "Implementation" – är en samling med relaterade portar, där varje port implementerar en specifik koppling av ett "abstract interface", en port = access till tjänsten



Beskrivning (WSDL)

- Huvudelementet i WSDL
 - **<types>** – beskriver datatyperna i meddelandet (vad)
 - **<message>** – definierar formatet som meddelandet har (vad)
 - **<portType>** – definierar en mängd av operationer
 - <operation> definierar en operation och dess in/ut meddelanden (vad)
 - **<binding>** – mappar en <portType> till ett konkret protokoll och dataformat (hur)
 - **<service>** – definierar en samling av relaterade portar, en <port> mappar en <binding> till en URL (var)



Upptäcka/Registrera (UDDI)

- Registrertjänst som är implementerad som en web-service i sig
 - SOAP-meddelanden används för att kommunicera med ett UDDI-register
- Mestadels används UDDI i utvecklingsfasen men kan användas under körning också
- Hanterar information
 - Tjänstetyp
 - Tjänstutövare
- Erbjuder mekanismer för att:
 - Kategorisera tjänster och utövare
 - Hitta en tjänst
 - Kryta till en tjänst

UDDI (registrering)

- En tjänstetyp definierad av en så kallad **tmodel** definierar en abstrakt tjänst
- **tmodel** tillhandahåller en referens till ett WSDL dokument som beskriver **vad delen** (abstract interface)
- Tjänstutövare registrerar sin verksamhet och alla tjänster som den erbjuder
 - För varje erbjuden tjänst så tillhandahåller den information som för bindningen till tjänsten, dvs en referens till WSDL dokumentet som beskriver **hur-delen**
- En verksamhet, tjänst eller tmodel som registreras kan kategoriseras enligt ett antal taxonomier fördefinierade eller utökningar till registret för att öka fokusering anpassad indelning och sökning

UDDI (Sökning)

- Sökning i ett register kan göras efter
 - En verksamhet som erbjuder en viss typ av tjänst
 - En tjänstetyp bland tjänsterna
- Taxonomierna kan användas för att göra frågor
- Från en tmodel för en tjänstetyp kan man komma åt
 - WSDL-beskrivningen av vad-delen, som kan användas/kompileras för att få ett klientgränssnitt till den tjänsten som kan accessa alla tjänster av denna typ
- Från bindningmönstret för en viss tjänst får man fram accessinformation och WSDL beskrivningen av binding informationen

UDDI (bindning)

- Statisk bindning
 - Underkompilering och runtime
 - Genom hur-delen kan en utvecklare kompilera ett konkret SOAP gränssnitt för en klient
 - Accessen kan bestämmas under körning
- Dynamisk bindning
 - En klient kan under körning dynamiskt kompilera var-delen (som innehåller hur-delen) för att generera bindningen



UDDI (dynamisk sökning)

- UDDI själv är en web-service
 - Förfrågningar från klienter under körning
 - Upptäcka tjänster
 - Lokalisera access till tjänster
 - Hämta en tjänst WSDL
 - Kryta till den



Andra mekanismer

- Direktaccess (man vet vart och hur ...)
- WS-inspection language
- P2P
- Index vs Register
- Sökmotor vs förmedlare

Typer av UDDI-register

- Privata
- B2B

Jobba med SOAP, WSDL & UDDI

- JAVA API
 - JWSDL – API för att jobba med WSDL
 - JAX-RPC – API för XML baserad RPC :-)
 - JAXM – API för att jobba med XML-meddelanden SOAP utan WSDL...
 - SAAJ – ägnad SOAP API
 - JAXR – API för att jobba med XML register typ UDDI
- Microsoft
 - .net Framework
 - SOAP toolkit

Implemetera WSA

- Standard XML verktyg
 - Jobbigt, svårt och tidskrävande
- Web-service plattformar
 - Utvecklingverktyg
 - "Runtime" SOAP server - tar hand om SOAP-meddelanden (tolkar, översätter, etc) till applikationsspråket
 - Administrationverktyg



- WASP
 - JAVA och C/C++
- J2EE
- .NET
 - Många språk
- AXIS
- IBM !!

