

Rapport från SOA & Cloud Symposium

Heta diskussioner i höstveckert Berlin



2010-10-20: Sven-Håkan Olsson

SOA LEVER Trots att begreppet SOA hamnat lite i bakvatten var årets SOA & Cloud symposium riktigt välbesökt och diskussionerna gick heta. Inte minst kom REST kontra SOAP och hur REST faktiskt används upp på tapeten.

SOA Symposium hölls just för tredje gången, denna gång i ett höstveckert Berlin.

Evenemanget var samorganiserat med Cloud Symposium, något som är naturligt eftersom nästan all molnanvändning förutsätter att ett antal kopplingar går att göra mellan interna system och de i molnet. Detta oavsett om ihopkopplingarna görs enligt SOA-principer eller med andra sorters integration. De här frågorna är förmodligen några av de alla viktigaste vid ett molninförande men tyvärr kommer detta ofta bort i debatten.

De två dagarna bjöd på många intressanta sessioner. Det var i själva verket svårt att välja eftersom det var så många bra ämnen som behandlades parallellt. En hel del besökare kom från tyskspråkiga länder men alla sessioner hölls på engelska.

Här kommer några spaningar och intryck som jag samlade på mig under dagarna i Berlin:

REST

En slutsats från dagarna är att den så kallade REST-tekniken har kommit för att stanna (se en.wikipedia.org/wiki/Representational_State_Transfer). Den är i sig mycket enklare och behändigare än den absolut vanligaste standarden för Web Services hittills, SOAP. Första bokstaven i SOAP stod tidigare för "simple", men eftersom de nyare standarderna WS-* är mycket komplexa har man gett upp och helt enkelt definierat bort att SOAP

skulle uttydas "simple". Numera ska SOAP endast vara ett egennamn, ingen förkortning. Klart är att en del debattörer har starka åsikter angående REST kontra SOAP.

Men det finns också en annan strid, den mellan förespråkarna för ortodoxa "truly RESTful services" och en användning av REST endast som ett extra behändigt överföringsprotokoll för SOA-anrop. Här har många personer ännu mer kraftfulla åsikter.

För egen del tycker jag att man kan se pragmatiskt på det här. Det uppstår egentligen fyra SOA-alternativ och vilket man beslutar sig för bör bero på vad man har för förutsättningar:

Enkel SOAP-användning

Välspritt, mycket interoperabelt, alla verktyg klarar det. Det finns dock vissa begränsningar man måste förstå.

Komplex SOAP- användning baserad på WS-*

Inte fullt så välspritt, inte alltid i praktiken interoperabelt, svårare att felsöka på grund av komplexitet. Man blir vanligen beroende av mellanprogramvara som i sig är komplex och kräver kunskap. Har dock många fler inbyggda säkerhetsfunktioner som man annars måste ordna på annat sätt.

REST som ersättare till SOAP för traditionell SOA-interaktion

Mycket enkelt, ändå kraftfullt. Säkerhet får lösas med traditionella webbt tekniker som måste överenskommas. Informationsbeskrivningar behöver tillföras.

Ortodox REST

Här handlar det om ett helt annat sätt att tänka, tjänstemodelleringen blir totalt annorlunda. REST har en extra potential då tjänster ska exponeras för en mycket stor användarskara som huvudsakligen behöver läsätkomst. Då man däremot baserar tjänster på gamla investeringar i verksamhetssystem kan det vara svårt att klara av ortodox REST.

BPM

En annan slutsats är att man på några håll har kommit långt med användningen av lösningar för BPM (Business Process Modeling, eller ibland Business Process Management. Visionen är att på hög nivå, i eller nära verksamheten, flexibelt kunna "programmera" en affärsprocess. På andra håll verkar man faktiskt inte kommit så långt alls eftersom det tyvärr finns utmaningar med denna tanke. Bilden är alltså splittrad.

Ett av flera problem med BPM är att man blir beroende av stora programpaket för både modellering och exekvering. Programvarorna är ofta kostsamma och kräver stor kunskap för att användas rätt. Det finns risk att bli inlåst i programpaketet trots vissa ansträngningar kring standardisering.

Ett vanligt mönster inom området utgör ett annat problem: sammansatta tjänster ("composite services"). Här finns stor risk att tappa bort data. Mitt eget föredrag på symposiet handlade om detta: *ACID transactions, so good but cannot be used in SOA, so what do we do?*. En kommande trendspaning planeras behandla det ämnet.

SOA

SOA fick för något år sen fick lite dåligt rykte eftersom ett antal "luftiga" SOA-projekt inte levde upp till förväntningarna. Men en tredje slutsats från symposiet är att det numera finns ett större självförtroende inom SOA-sfären. De värsta avarterna har hunnit förkastas. Min egen åsikt är att precis som vid problemen med BPR (Business Process Reengineering) i början på 90-talet så handlar det även idag om att klara de organisatoriska förändringarna, inte bara teknikpysslet.

Flera diskussioner under dagarna handlade också om i vilken ordning man helst bör modellera. Ska man börja helt förutsättningslöst från verksamhetsidan och sedan låta det bli "lagen", eller ska man utgå från vad som finns i existerande system? I första fallet är risken att systemkostnaden blir enorm, i andra fallet att verksamheten inte kan utvecklas optimalt. Som vanligt är det förmodligen både och som gäller. Jag har därför i många år förespråkade begreppet *kravdialog*.

Som avslutning, äntligen såg jag iPad:s och konkurrerande plattor användas till något "nyttigt", annat än för underhållning. Att skriva anteckningar under föreläsningarna och kanske rita små figurer verkade fungera bra.

Läs mer här: www.soasympoium.com, www.cloudsymposium.com och en.wikipedia.org/wiki/Representational_State_Transfer (dvs REST).



Sven-Håkan Olsson är en fristående konsult som särskilt arbetar med att kombinera verksamhetsnytta med teknikhöjd. Han har en lång karriär sedan 70-talet som it-konsult (it-arkitektur, systemdesign, programmering, reviewer, utredningar, kursledning). Sven-Håkan är också medgrundare av Know IT och var dess teknikchef 1990-2003. Han utsågs till en av "Sveriges topputvecklare" av Computer Sweden 2008 och 2010.

Sven-Håkan håller regelbundet kurser åt Dataföreningen Kompetens. Läs gärna mer på hans blogg www.definitivus.se.

[Sven-Håkan Olsson](#)