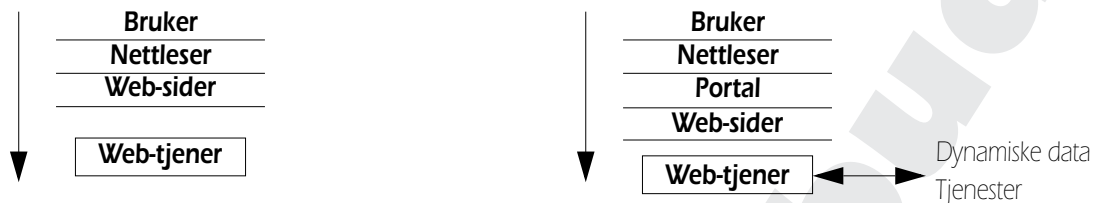




Yahoo.com ansees som verdens første portal. Vi ser også at et raskt voksende antall portaler inkluderer muligheter for personlig innlogging og individuelle tilpasninger. Mens dette kan være nyttig og effektivt, er det ingen kvalifikasjon for bruken av begrepet 'portal'.



**Figur 2** Portaler er ikke det samme som Web-sider, men et nytt nivå introdusert mellom brukeren og informasjonsholdige Web/sider. For brukeren er portalen inngangsport og innholdsfortegnelse på høyt nivå for ulike samlinger av informasjon og i voksende grad tjenester.

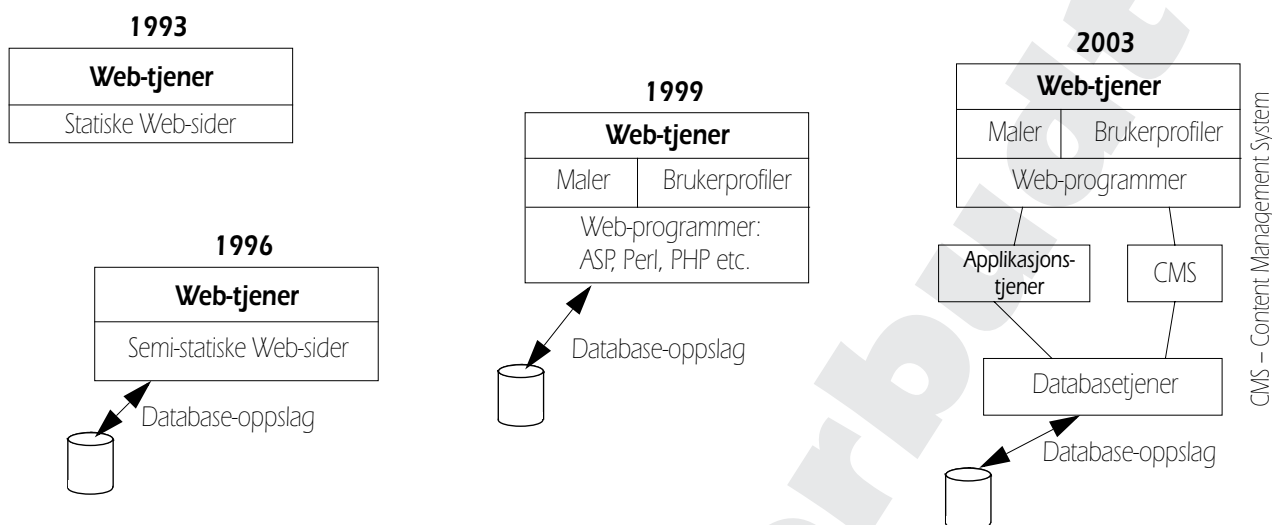
Kortvarianten av definisjonen blir dermed at portaler er introdusert som et nytt nivå mellom brukere og Web-sider – og et voksende antall andre elementer som vi skal komme tilbake til nedenfor. På norsk blir portalen 'innholdsfortegnelsen' mens Web-sidene blir selve 'innholdet'.

### ... og fra tjener til tjenester

På samme måte som funksjonaliteten på klientsiden har utviklet seg dramatisk – med utvidelser av JavaScript og tilveksten av JScript (Microsofts JavaScript-klone), Java, ActiveX og .NET, har tjenersiden bokstavelig talt eksplodert – i flere retninger. Utviklingen representerer selve fundamentet for det som i dag kalles Web-tjenester, og som vil bli en viktig bærebjelke i løsninger av alle tenkelige slag de neste 5 årene. Vi kommer tilbake til betydningen av Web-tjenester nedenfor.

Figur 3 illustrerer progresjonen fra de første Web-tjenerne (ca. 1993) til dagens situasjon. Illustrasjonen fremhever det faktum at nye systemarkitekturer må til for å betjene de behovene som er i ferd med å oppstå, hvilket var et hovedpoeng i analysen av systemarkitekturer i forrige utgave (se figur 4). Dette forholdet er av vesentlig betydning når vår fremtidige IT-infrastruktur skal planlegges.

Arkitekturene som har utviklet seg fra 1999 og frem til i dag, representerer et spennende og interessant brudd med homogeniserings-tankegangen som har preget deler av IT-utviklingen de siste årene. Nok en gang har erfaringer vist at mens universalsystemer ser ideelle ut på papiret, er de sjelden optimale i praksis. Misforståelsen har ligget i at standardisering og homogenisering forutsetter bruk av universalsystemer på tjenersiden, hvilket er å angripe problemstillingen i gal ende. Homogenisering er viktig der det er mulig, og spesielt der hvor antall enheter er stort. Derfor må utfordringen først angripes fra klientsiden. På tjenersiden er skreddersøm og optimaliseringer i voksende grad nødvendige for å betjene et belastningsbilde som skiller seg radikalt fra hva vi har vært vant med siden midt på 90-tallet.



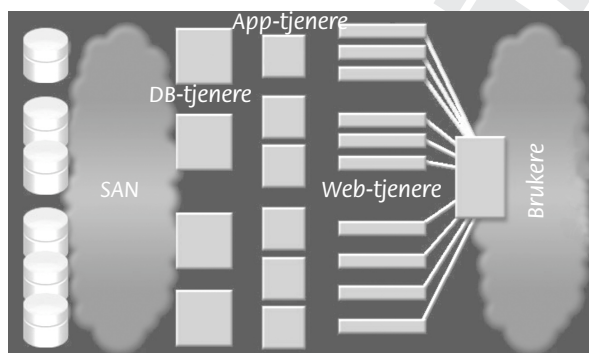
**Figur 3** Utviklingen på tjenersiden fra WWWs spede begynnelse i 1993 og frem til i dag er påtagelig, men begynner først nå å få konsekvenser for de underliggende systemarkitekturene.

### Valgets kval

Hvordan velger vi så en portal-løsning? Spørsmålet dukker opp med voksende hyppighet fra klienter og lesere, og har naturligvis intet fasit-svar. Det viktigste kriteriet er å vite hva vi vil – hvilke behov som skal dekkes, hvilket foranlediger et krav om å vite hva som er mulig. Og her stopper det for de fleste. De overlater til leverandørene å fortelle hva som er mulig. Deretter velges den leverandøren som skaper mest tillit – i stedet for den løsningen som dekker behovene best. Det finnes mer enn et dusin attraktive kommersielle alternativer på markedet – i tillegg til en håndfull Open Source varianter som slett ikke er til å kimse av.<sup>1</sup> Hvem som skal gjøre jobben med installasjon, tilpasning, drift og

løpende oppdateringer er nøkkelparametre for valget: Å utnytte eksisterende kunnskap og erfaring er alltid en fordel dersom det lar seg forene med kravspesifikasjonen forøvrig.

Et annet viktig spørsmål i forbindelse med valg av produkt og verktøy er integrasjon kontra modularisering. De store leverandørene ønsker å ha kontroll over hele prosessen fra A til Å, hvilket har praktiske fordeler både under utvikling og drift, men samtidig fører til en låsing som i mange tilfeller er mindre ønskelig på sikt. Egen kompetanse og langsiktige planer er avgjørende for hvilket valg som er optimalt. Å velge en modulær løsning som gir fremtidig fleksio-



**Figur 4** En lagdelt tjener-arkitektur gir både fleksibilitet, robusthet og god økonomi for betjening av brukere som i raskt voksende grad kommer inn via nettlesere og HTTP.

<sup>1</sup> En spesielt interessant variant i den forbindelse er Zope, som ikke bare har lang fartstid og et bredt funksjonsspekter å spille på, men også besitter den modulære strukturen som gjør det mulig å videreutvikle portal-arkitekturen også etter at den er satt i drift. Team Mellviks Web-tjener er i disse dager under omarbeiding til portal med Zope som utgangspunkt, med Zope CMS (CONTENT MANAGEMENT SYSTEM) og Plone. Sistnevnte er rammeverk for brukergrensesnitt og styringsgrensesnitt. [www.zope.org, www.plone.org]

## Inn med Web-tjenester

### Web-services (protokoller)

|  |
|--|
| UDDI (discovery)<br>WSDL (description)   |
| SOAP<br>(Remote Service Call)            |
| HTTP<br>(Transport Application Protocol) |
| TCP/IP                                   |

Web-tjenester er mekanismer som gjør det mulig å koble sammen vilkårlige tjenester og applikasjoner uten å 'grave' i den enkelte applikasjons programmeringsgrensesnitt. Standardiseringen og forenklingen åpner et formidabelt potensiale for samspill mellom løsninger.

**UDDI** – Universal Description, Discovery and Integration

**WSDL** – Web Services Description Language

**SOAP** – Simple Object Access Protocol

**HTTP** – HyperText Transport Protocol

bilitet med hensyn til bruk av teknologi og leverandører, er alltid å foretrekke dersom det lar seg gjøre i praksis. Likeledes er valget av grunnlagsteknologi kritisk for hvilke veier som ligger åpne fremover: .NET representerer en låsing til Microsoft på godt og vondt, mens Java gir et spekter av leverandører og teknologier å velge fra.

Som vi var inne på innledningsvis, er ei heller uttrykket 'Web-tjenester' så entydig som vi kunne ønske. I mer enn 2 år har de største markedsaktørene kranglet om hvem som har den korrekte definisjonen, hvem som var først ute og ikke minst hvem som har de beste svarene på utfordringene som venter. Selv World Wide Web Consortium (W3C, [www.w3c.org](http://www.w3c.org)),<sup>2</sup> som utarbeider og godkjenner de grunnleggende standardene på området, har ikke maktet å eliminere forvirringen – som delvis er forårsaket av selve begrepet: Web-tjenester er ikke tjenester levert via nettleser til brukere, men standard-mekanismer som gjør det mulig å 'lime sammen' nye og gamle løsninger med utgangspunkt i Web-teknologi og eksisterende infrastruktur. At disse mekanismene også naturlig kan betegnes som tjenester, er en del av forvirringen. Vi snakker om tjenester på ulike nivåer, og Web-tjenester er aldri direkte synlige for sluttbrukere.

En kortfattet definisjon av Web-tjenester kan formuleres som følger: *Web-tjenester er mekanismer som via en håndfull uavhengige protokoller besørger oppslag, forespørsler og dataoverføring via HTTP mellom løst koblede noder i IP-baserte nettverk.* Punktene nedenfor gir mer kjøtt på beinet:

- ✓ Årsaken til at 'Web' er en del av uttrykket er at HTTP benyttes som transportprotokoll, hvilket gjør det enkelt og naturlig å implementere de nye mekanismene innenfor rammene av eksisterende Web-tjenere.
- ✓ Mens HTTP er transport-protokollen, er det definert en gruppe nye protokoller som tar vare på hovedfunksjonene (1) søking/oppslag (UDDI), (2) tjenestespesifikasjon (WSDL) og (3) dataoverføring (SOAP) – se margrammen.
- ✓ Mens HTTP (og i sin tur TCP/IP) tar vare på datatransporten, er XML fundamentet for all datautveksling.

Et banalt, men nyttig eksempel på hvordan dette fungerer i praksis er en applikasjon som sender avgårde et postnummer til en tjeneste, for deretter å få tilbake en oppdatert temperatur på det aktuelle stedet. Selv for en så enkel transaksjon ser vi at det er en betydelig mengde elementer som for det første må finnes, og for det andre må kunne brukes via enkle standardmekanismer – for eksempel:

- ✓ Hvordan finner vi en slik tjeneste?
- ✓ Hvordan velger vi dersom det finnes flere av dem?

<sup>2</sup> W3C er ikke alene om å arbeide med standarder på området. To andre enheter som det er verdt å kjenne til, er OASIS (ORGANIZATION FOR THE ADVANCEMENT OF STRUCTURED INFORMATION STANDARDS – [www.oasis-open.org](http://www.oasis-open.org)) og WS-I (WEB SERVICES INTEROPERABILITY ORGANIZATION – [www.ws-i.org](http://www.ws-i.org)).

- ✓ Hvordan formuleres forespørselen (innhold, format etc.)?
- ✓ Hvilken annen informasjon kan vi hente fra samme sted i samme transaksjon?
- ✓ Hvilket format har dataene som kommer tilbake?

Innenfor snevre, kontrollerbare rammer har vi hatt svarene tilgjengelige i årevis. Såkalte EAI (*Enterprise Application Integration*) verktøy har vært i bruk i 10 år, og har vært nyttige om aldri så kompliserte og snevre i sitt nedslagsfelt. Utfordringen er å forenkle og generalisere det hele, å utnytte standardisert Internett-teknologi til å sy sammen løsninger på et høyere nivå. Måten portaler og Web-tjenere samarbeider på fremstår som en attraktiv modell i så måte, og er samtidig et glimrende utgangspunkt å bygge videre på.

Med universelle svar på spørsmålene ovenfor, står vi potensielt overfor en løsning på den største utfordringen IT-verden har slitt med de siste 20 årene: Effektivt samspill og datautveksling mellom applikasjoner. Faktum er at overvekten av alle programmer som utvikles i dag, på en eller annen måte er involvert i datakonvertering på ulike nivåer. Den kolossale interessen for Web-tjenester har sin årsak i nettopp dette forholdet. Dersom det virkelig er mulig å forenkle og standardisere mekanismene og få resultatet til å fungere i praksis, har verden gjort et utviklingsmessig kvantesprang. Og utgangspunktet er unektelig det beste. TCP/IP er allesteds nærværende, HTTP er det samme – og dessuten både enkel og rimelig effektiv.

### 3 byggeklosser

Konseptet *Web-services* består av 3 distinkte byggeklosser:

- ✓ En standardisert katalogtjeneste med oversikt over tilgjengelige tjenester (UDDI) kombinert med en beskrivelses-standard for tjenestene (WSDL).
- ✓ En mekanisme som gir tilgang til tjenestene (forespørsler og svar – SOAP).<sup>3</sup>
- ✓ En standard transportmekanisme for transaksjonene (HTTP).

Metoden tilsvarende hvordan vi ofte benytter Internettet i dag. Vi går først til en søketjeneste med et søkebegrep. Svaret gir oss en oversikt over 'treff' – som kan være tilfredsstillende etter første runde eller kreve justering av kriteriene. Resultatet er grunnlag for ett (eller i noen tilfeller flere) valg: Vi velger et nettsted og går dit for å fullføre transaksjonen, eller – dersom vi fortsatt ikke er tilfreds – et trinn tilbake og forsøker neste.

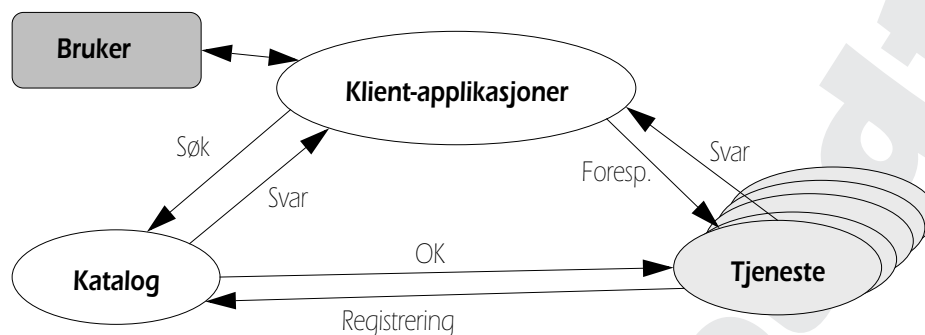
For Web-tjenester er prøve-og-feile delen av transaksjonen sjelden akseptabel. Derfor må tjenestebeskrivelsen (WSDL) være tilstrekkelig utfyllende til at vi er sikre på å få tilbake det vi forventer når valget gjø-

<sup>3</sup> Det bidrar ytterligere til forvirringen omkring Web-tjenester at standardene vi nevner her ikke er de eneste som kan brukes for å implementere tjenestene. Det er funksjonaliteten som til sammen gir en Web-tjeneste, ikke at akkurat disse standardene benyttes. SMTP kan brukes i stedet for HTTP, HTML kan brukes i stedet for SOAP etc. På den andre siden er det også et faktum at nettopp UDDI, WSDL, SOAP og HTTP normalt er underforstått når 'Web-tjenester' omtales.

Katalogtjeneste  
Beskrivelser

Meldings-  
protokoller

Transport-  
protokoller



**Figur 5** En forenklet skisse over hvordan Web-tjenester er limet i sammenkoblingen av applikasjoner. De tilgjengelige tjenestene registreres i en katalog som benyttes til oppslag og søking av klientene. Når en klient har behov for en tjeneste, foretar den først et søk, og utfører deretter én eller flere transaksjoner mot den valgte tjenesten.

res og transaksjonen utføres. I og med at byggeklossene er basert på XML, gir de i utgangspunktet fullstendig plattformuavhengighet. Tjenestebeskrivelsen (WDSL) forteller i sin helhet hvilke data som sendes i den ene og den andre retningen, og deres formater. Hardware, operativsystemer og andre infrastruktureddetaljer kan påvirke responstider, men ikke resultatene.

En leverandør av for eksempel digitale bilder vil – etter å ha registrert sine tjenester i en offentlig katalogtjeneste – være tilgjengelig for *business* – i prinsippet fra hele verden, og gjennomføre transaksjoner raskt og effektivt via SOAP-protokollen. Tilsvarende kan gjøres i dag via nettleseren, men langt mer tungvint – spesielt dersom det skal søkes hos flere leverandører. Videre har alle leverandører sin måte å søke, vise frem og gjennomføre en eventuell handel på.

Mens Web-tjenester blir et attraktivt grunnlag for utvikling av nye tjenester, representerer muligheten til sammenkobling av eksisterende løsninger og applikasjoner – uten tanke på alder, opprinnelse, plattform eller dataformater, konseptets mest attraktive side. Verden har sett utallige forsøk på å løse nettopp disse utfordringene de siste 20 årene. Når ingen har vært vellykkede utover smale nisjer og spesifikke scenarier, er hovedårsaken at de har manglet nettopp det Web-tjenester er sterkest på: Uavhengighet – av leverandør, plattform, plassering og formater.

Sagt på en annen måte gir *Web-services* mulighet til å virtualisere datatjenester til et nivå som hittil knapt har vært en drøm i de fleste miljøer. Dermed dukker forholdet vi var inne på ovenfor, opp nok en gang: Vi kan homogenisere og standardisere på klientsiden og i nettverket – der det åpenbart har størst effekt, samtidig med at vi på tjenersiden utnytter eksisterende skreddersøm og etablerer ny etter behov. Standardiseringen på mellomvare-nivå eliminerer behovet for bruk av proprietære programmeringsgrensesnitt som ikke bare forutsetter skreddersøm for hvert produkt eller tjeneste, men som også i mange tilfeller er lisensbelagte og dermed kostbare.

Hvor stor avstanden er fra drøm til virkelighet kommer vi tilbake til nedenfor. Allerede nå har vi imidlertid sett nok av potensialet i Web-

#### Office mot Web-tjenester

Mens vi så langt kun har sett begynnelsen av de forandringene Web-tjenester lover å forårsake, mangler vi ikke interessante eksempler. Et av dem kommer fra Houston Texas, der kommunen tidlig i år valgte å fase ut Office til fordel for en løsning basert på Web-tjenester fra en lokal, relativt nystartet programvare-leverandør. Konseptet har Web-tjenester i bunnen og leverer grunnleggende Office-funksjoner og -kompatibilitet, uten å konkurrere med Microsoft – sier leverandøren SimDesk Technologies [www.simdesk.com]. Videre er løsningen utstyrsuavhengig, og fungerer like godt på mobiltelefoner som på fete PCer. Også flere andre miljøer i offentlig sektor i USA tester produktet, og vi kan ikke la være å stille spørsmålet: Hva er medaljens bakside? Er ikke dette for godt til å være sant? Nettopp derfor kommer vi tilbake til produktet i spalten 'Godbiter' i neste utgave av Mellvik-Rapporten.

tjenester til å forstå hvorfor både leverandørsiden og markedet i sin alminnelighet med stor iver har kastet seg over teknologien. Ikke minst har Microsoft og Sun Microsystems satt sitt preg på jippen: Sentrert rundt henholdsvis .NET og Java, har begge omfattende produktsamlinger og strategier knyttet opp mot nettopp *Web-services*. Disse teknologiene blir diskutert i artikkelen på side 13.

### Standarder og kaos

De tre standardene vi har presentert ovenfor, representerer et fundament, men kun en spe begynnelse til hva som skal til for å realisere Web-tjenester i stor skala. På nivåene over – og i noen grad ved siden av – foreligger det mer eller mindre ferdige forslag til ytterligere 30+ standarder. Tradisjonen tro krangler leverandørene så busta fyker om hva som er bra og ikke bra i den forbindelse. Her er det kun unntaksvis reelle faglige eller tekniske problemstillinger som diskuteres, men forhold knyttet til hva som ligger nærmest opp til den enkelte leverandørs eksisterende teknologier og preferanser forøvrig. Fra tid til annen kan det synes som om det er viktigere at konkurrentene ikke får rett enn at resultatet blir bra. Interesseorganisasjonene – W3C, WS-I og OASIS – forsøker etter beste evne å være nøytrale pådrivere, men har ved flere anledninger møtt veggen, med den følge at standardiseringsarbeidet midlertidig har stoppet opp.

Potensialet er imidlertid for stort og interessene for sterke til at noen er tjent med at prosessene stopper. Dessuten – og heldigvis – er situasjonen slik at ingen leverandør er vesentlig sterkere enn de andre, hvilket medfører at reelle konkurransekrefter får virke uhemmet. Derfor har vi sett ikke bare vesentlige fremskritt med hensyn til standarder det siste året, men også glimrende eksempler på praktisk bruk av Web-tjenester.

#### Amazon.com, Google og eBay omfavner Web-tjenester

Mens tallrike viktige standarder fortsatt er under utvikling, har nettstedet Amazon.com omfavnet *Web-services* med begge armer. Selskapet har en enorm produkt-database og et usedvanlig velutviklet apparat for nettsalg, og ser stort potensiale i at andre får tilgang til disse tjenestene. Dokumentasjon, utviklingsverktøy (*tool-kits*) og tallrike praktiske eksempler skal hjelpe utviklingsmiljøer å ta i bruk Amazons Web-tjenester – og strategien har lyktes. Interessen har vært formidabel, og har sørget for at også selskapets europeiske avgrenning støtter tjenestene.

Store organisasjoner integrerer Amazons Web-tjenester i sine interne innkjøps-portaler, og effektiviserer på den måten innkjøp av langt mer enn bøker.

Den omfattende utviklingsstøtten er nødvendig gjort av det faktum at en rekke standarder fortsatt er mangelfullt utviklet. "Dette er meget viktig for oss", sier Amazon.com. "Vi tilpasser oss standardene så snart de blir ratifisert. I mellomtiden gjør vi det som skal til for at tjenestene skal fungere og la seg integrere." Her er naturligvis også PR-effekten selskapet får i tekniske miljøer et viktig element.

Andre gode eksempler er søketjenesten Google, som legger forholdene til rette for å fungere som foretrukket søkemotor for portaler i sin alminnelighet – via Web-tjenester, og auksjonsstedet eBay med tilsvarende tankegang. Felles for alle tre er at de ser potensiale for nye inntektskilder – noe ingen av dem har klart å realisere enda, men som like fullt kan sannsynliggjøres på sikt.

Eksemplet fra Houston, Texas (se margrammen på forrige side) er spennende fordi dagligdagse problemstillinger angripes fra en ny vinkel, mens Amazon.com (se ramme) demonstrerer potensialet for forenkling, tilgjengelighet og nye forretningsmuligheter. Andre nærliggende eksempler er Web-tjenester som leverer vær- og temperaturinformasjon for vilkårlige steder på kloden, eller løpende børs- og nyhets-tickere.

Hvilke standarder er det så som mangler, hvilke områder er det som dekkes av disse mer enn 30 standardene som er under utvikling eller nylig ratifiserte? Sikkerhet er et innlysende felt i den forbindelse: Gjensidig identifikasjon og autentisering av klienter og tjenere, samt transportsikring og transaksjons-pålitelighet. Web-tjenester i sin almin-

nelighet blir meningsløse uten pålitelige mekanismer for å beskytte både partene og innholdet i transaksjonene.

Følgende punkter dekker de viktigste, om ikke alle områdene som berøres av disse standardene:

- ✓ **Sikkerhet:** Flere standarder og spesifikasjoner med et fundament som kalles *WS-Security* (OASIS).
- ✓ **Arbeidsflyt** – både W3C og OASIS har aktive prosesser på området: *Web Services Business Process Workflow* (OASIS), *Web Services Choreography Interface* (W3C).
- ✓ **Pålitelig meldingsutveksling:** Transaksjons-integritet og leveringspålitelighet. *WS-Integrity* klargjøres i disse dager av OASIS,<sup>4</sup> mens Microsoft/IBM/BEA arbeider med et alternativt forslag som kalles *WS-ReliableMessaging*.
- ✓ **Styring:** En arbeidsgruppe i OASIS skal fremlegge forslag til *Web Services Distributed Management Framework* tidlig neste år.

De største og mest aktive aktørene fra leverandørsiden – ved siden av Microsoft og Sun Microsystems – er Oracle, IBM, BEA og SAP. Samtlige er aktive i utviklingen av standarder og tilhørende spesifikasjoner. Spesielt har kombinasjonen Microsoft/IBM utmerket seg – med høyt aktivitetsnivå og allokering av store ressurser. Her er det viktig å smi mens jernet er varmt, og samtlige aktører vet at aktiv deltagelse er den eneste måten å påvirke resultatet på.

## Konklusjon

Hvordan kommer vi så til det forfjettede land? Ligner ikke mange av disse drømmene på noe vi har sett før, luftige lovnader med beskjedne forankring i virkeligheten fra ordrekåte leverandører? Her er det uten tvil mye skrik og enn så lenge lite ull. Eksempler fra virkeligheten viser imidlertid at Web-tjenester allerede har passert de første milepælene

på veien fra idé til virkelighet. Samtidig er det vår oppgave å være realistiske og å sette tingene i et rimelig tidsperspektiv: Er Web-tjenester noe vi må forholde oss til på kort sikt, altså innenfor den nærmeste 18-måneders perioden?

For oss som er i bransjen – på leverandør- eller kundesiden – er svaret ubetinget ja. Om Web-tjenester ikke kan gjøre underverker for oss i denne perioden, er potensia-

### Med Microsoft som drivkraft

Microsoft har helt siden den spede begynnelse vært en sentral drivkraft i utviklingen av standardene bak Web-tjenester. Dette engasjementet har overrasket mange i og med at flere av selskapets egne teknologier, herunder COM og DCOM, undermineres av de nye standardene. Omfavning av åpne standarder på bekostning av egne teknologier er ikke en vanlig kost fra den kanten. Sammenhengen er imidlertid ikke fjern. Heller ikke Microsoft har full kontroll over markedet og utviklingen, og må fra tid til annen velge mellom onder. Da det etterhvert ble klart at Java ikke lot seg utmanøvrere, bestemte Microsoft seg for å satse på en hest som langt fra var ideell, men som kunne stoppe Suns Java-baserte teknologier fra å bli totalt dominante i et voksende marked.

Denne hesten het *Web Services*. Med sitt XML-grunnlag, en standard Microsoft allerede hadde lagt ned betydelig innsats i, kunne tilsvarende Java-baserte mekanismer, sentrert rundt JRI – Java Remote Invocation, stoppes og noe av utviklingen bringes tilbake under kontroll. Denne strategien lykkes, og Web Services spesifikasjonene er tilstrekkelig løse i kantene til at det er lett å lage dialekter etter behov. Så vil tiden vise om Microsoft følger samme mønster som tidligere, eller velger å holde seg til de offisielle standardene – som selskapet selv har vært med på å utvikle.

<sup>4</sup> Denne spesifikasjonen er fundamentert på et utkast fra Sun, Oracle, Fujitsu, Hitachi, NEC og Sonic Software tidligere i år.

let, aktivitetene og utviklingshastigheten mer enn store nok til at dette er noe vi må ta hensyn til både i forbindelse med utviklingsprosjekter og generell planlegging. For eksempel er det knapt hensiktsmessig å velge tradisjonelle, kompliserte og helproprietære EIA-plattformer – for eksempel Microsoft BizTalk – dersom målet kan nås via Web-tjenester.

Som vi var inne på innledningsvis, er det langt mellom eksemplene på at Web-tjenester muliggjør noe som tidligere var umulig. Det er måten applikasjoner og tjenester kobles sammen på som er annerledes: Enklere, mer standardisert, mer plattformuavhengig og med mindre rigide leverandørbindinger. Valg av verktøy er åpenbart kritisk i den forbindelse. Om standardisering og konseptuell uavhengighet er aldri så bra, er sjansen stor for at verktøyene vi velger, reduserer frihetsgraden dramatisk. Dette forholdet representerer derfor et viktig kontrollpunkt når verktøy og leverandører skal velges. Likeledes, og like viktig er det å ha i mente at de standardene som finnes og er etablerte, kun representerer begynnelsen. Vi snakker om et mål i bevegelse – uenkelig en ulempe og samtidig det vi kaller en *fact of life*. “Det er en kolossal utfordring å få alle bitene på plass – vi bygger fremtiden, og det tar tid”, sa Microsofts Felipe Cabrera nylig på O’Reillys *Emerging Technologies Conference*.

Mens Microsoft fokuserer på kompleksiteten og utfordringene, har andre leverandører valgt den motsatte angrepsvinkel – enkelhet og umiddelbare muligheter. Slik skal det være – drivkreftene og idéskapningen sørger for å akselerere utviklingen, blant annet ved å skaffe praktiske erfaringer som i sin tur påvirker standardiseringsprosessen.

*Web-services* er inne på radarskjermen, og kommer til å forårsake betydelig positiv oppmerksomhet i markedet de neste 12 månedene – et tema vi holder kontinuerlig under oppsikt og kommer tilbake til i Mellvik-Rapporten allerede 1. halvår neste år. ■