

Analyse, arkitektur og design av nettverk

Dette er 7. artikkel i en serie med fokus på bygging av moderne lokalnett: Behovsanalyse, krav, komponenter, innhold, design og styring. Serien er praktisk orientert på den måten at den gir råd og vink med hensyn til hvordan problemstillingene kan angripes for å komme frem til optimale resultater på kortest mulig tid.

Nettverksarkitektur

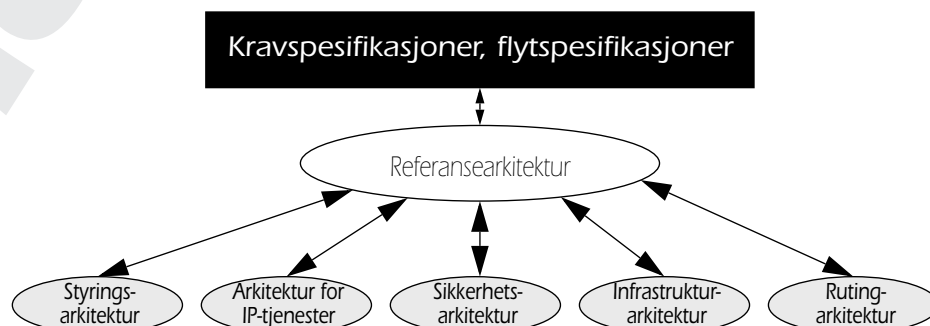
Vi har lagt bak oss analysefasen, og er godt forberedt på en etter de fleste oppfatning langt mer spennende og konkret oppgave: Nettverksarkitekturen – den overordnede planen som skal ligge til grunn for siste fase: Praktisk nettverksdesign.

Nettverksarkitektur – kan det virkelig være nødvendig? Det kan da ikke være SÅ vanskelig! Vanskelighetsgraden kommer naturligvis an på nettverkets omfang og kompleksitet, men det nettverk finnes knapt – utenfor gutterommet – som ikke har nytte av en arkitektur. Det samme kan sies for systemer, løsninger og sikkerhet – for å nevne noen eksempler.

Vi har ved tidligere anledninger benyttet følgende tidløse definisjon på nettverksarkitektur – en stadig like nyttig påminnelse om både hva vi nå gir oss i kast med, og hva en arkitektur ikke er:

En nettverksarkitektur er en overordnet målsetting og et sett spesifikasjoner som nettverket skal bygges rundt. I tillegg til overordnede faktorer må arkitekturen angi løsninger og/eller løsningsalternativer som dekker tre hovedområder: Fysisk arkitektur (konnektivitet/kabling), interoperabilitet (protokoller og tjenester), og systemer/prosedyrer for overvåking, styring og kontroll av nettverket. Arkitekturen inneholder ikke implementasjonsdetaljer som dimensjoner, båndbredder eller plasseringer av utstyr.

En annen viktig observasjon er at nettverksarkitekturen – i likhet med andre IT-relaterte arkitekturer – er under kontinuerlig utvikling. På samme måte som nettverket den beskriver, må arkitekturen kontinuerlig tilpasses virkeligheten – behov, muligheter, teknologi og fysiske forhold. Det betyr at en 'ferdig' arkitektur aldri hører hjemme i skuffen, men bør gjøres til et levende dokument som styrer planer og utvikling på infrastrukturens side.



Figur 5 En arkitektur er sammensatt av komponenter som i sin tur gjerne er arkitekturer i seg selv – for systemer, for sikkerhet og så videre. Detaljeringsgraden styres av behovene og påvirkes sterkt av infrastrukturens kompleksitet.

Figur 7, som er hentet fra seriens første artikkel, minner oss om arkitekturens plassering i prosessen: Den gir en overordnet eller 'høynivå' beskrivelse av nettverket, dets hovedkomponenter og sammenhengen mellom disse.

Med utgangspunkt i behovene – kravspesifikasjonene fra systemer, løsninger og brukere som vi har avdekket i analyseprosessen, skal arkitekturen fortelle hvordan enkeltelementene i infrastrukturen bør settes sammen og samspille for å dekke krav og behov. Den deles i hoveddeler (komponenter eller kapitler) etter behov, for eksempel som vist i figur 5. Vi kommer tilbake til disse hovedkapitlene nedenfor og i påfølgende artikler.

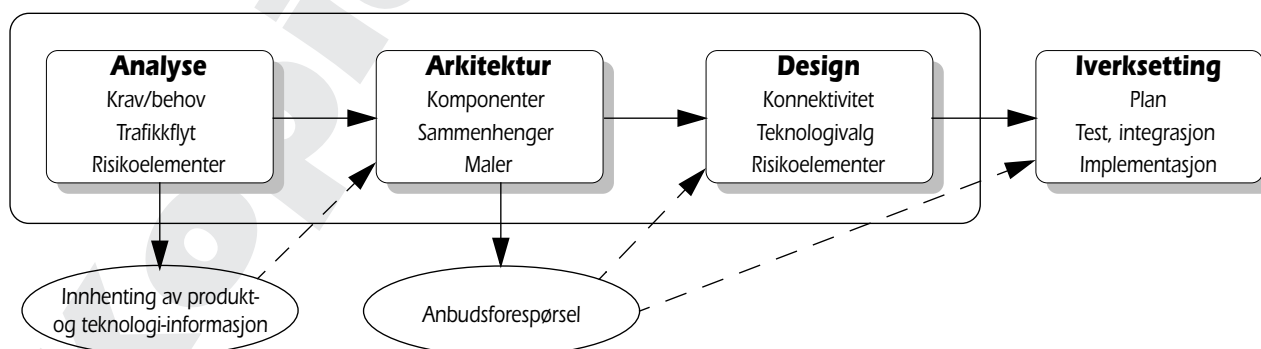
Erfaringsmessig er grensen mellom arkitektur og design i beste fall ullen, med tilhørende fare for å havne på gal side underveis i prosessen. Bruk oppstillingen i figur 6 som kontroll – i kombinasjon med praktiske hensyn og sunn fornuft.

Arkitektur		Design
Bred	← Omfang →	Fokusert
Generell	← Detaljeringsgrad →	Dyp
Sammenhenger	← Beskrivelse →	Teknologier
Uavhengig	← Fysisk/geografisk →	Avhengig

Figur 6 Grensen mellom design og arkitektur er ikke distinkt. En nyttig regel for å trekke riktige skillelinjer er å holde alt som er spesifikt (leverandør, modell, megabits, gigabytes, koordinater) i design, mens det generelle ('ruter', 'svitsj', 'linje', 'tjenester') hører hjemme i arkitekturen. En annen måte å uttrykke det samme på er å se på arkitekturen som en prinsippskisse, mens design skal kunne brukes til kostnadskalkulasjoner og innkjøp.

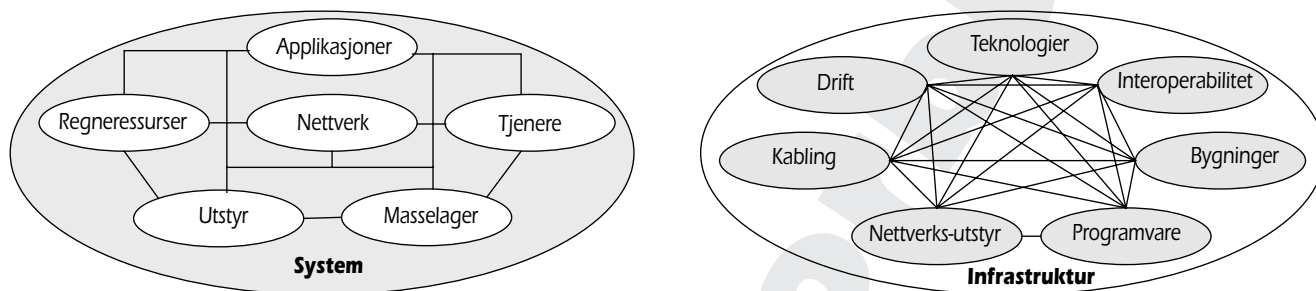
Arkitekturens hovedkomponenter

Som skissert i figur 5, består arkitekturen av en rekke underkomponenter – for eksempel systemarkitektur og infrastruktur-arkitektur. I sin tur kan disse deles inn i enkelt-komponenter eller logiske grupper – hvilket i noen tilfeller er nødvendig, i andre tilfeller overflødig. Figur 8 viser hvordan komponentoversikten kan ta seg ut for to av 'underarkitekturene'. Samtidig ser vi et typisk eksempel på høne-og-egg proble-



Figur 7 Et nettverksprosjekt består av fire hovedfaser. De tre første blir behandlet i denne artikkelserien. Figuren er hentet fra første artikkel i serien (se Mellvik-Rapporten nr. 100).

matikk: Systemarkitekturen er en del av nettverksarkitekturen, og inkluderer samtidig deler av den samme arkitekturen. Deres innbyrdes avhengighet er total, den ene kan ikke eksistere uten den andre. Med akademisk språk kan vi si at arkitekturen er rekursiv, hvilket først og fremst understreker avhengighetsforholdet. Nettverksarkitekturen skal betjene og muliggjøre systemarkitekturen, og blir dermed en del av den – på samme måte som veinettet betjener og blir en del av et boligområde.



Figur 8 En referansearkitektur består av komponenter som kan 'dekomponeres' videre til den grad av detaljering som er nyttig eller nødvendig.

Referanse-arkitektur

Kjært barn har mange navn. For å forsikre oss mot misforståelser benytter vi i denne sammenhengen begrepet referanse-arkitektur i stedet for nettverksarkitektur. Den rekursive egenskapen vi påpekte ovenfor, sørger for at 'nettverksarkitektur' ikke lenger er entydig.

Referanse-arkitekturen beskriver helheten – det sammensatte nettverket med samtlige komponenter. Mens alle komponentene i prinsippet er like viktige, vil omstendighetene være avgjørende for hvilke som faktisk finnes. Gjennomgangen nedenfor fokuserer på sentrale komponenter som alltid er med – i en eller annen form:

- ✓ IP-tjenester
- ✓ Sikkerhet
- ✓ Styring og drift
- ✓ Adressering og ruting
- ✓ Infrastruktur

I utarbeidelsen av arkitekturen er det en målsetting at vi ikke bare identifiserer komponentene, men også finner og forstår sammenhengen og samspillet mellom dem: Avhengigheter, begrensninger, avveininger som er gjort eller må gjøres og så videre. I denne og de neste artiklene skal vi konsentrere oss om nettopp dette. Vi begynner med en kort introduksjon av hovedkomponentene, og fortsetter med en mer detaljert gjennomgang av innholdet og eksempler på hvordan vi arbeider oss frem mot resultatet – som blant annet skal sørge for at vi ikke blir stilt overfor ubehagelige overraskelser når nettverket til slutt skal designes. Mer eller mindre åpenbare sammenhenger, samspill og konsekvenser er nøkkelfaktorer i den forbindelse.

IP-tjenester: Komponenten omfatter mer enn IP, og betegnelsen signaliserer først og fremst at vi har med protokoller å gjøre. Herunder havner forhold og mekanismer som:

- ✓ Prioritering, allokering av ressurser og kontroll (QoS) – på IP-nivå og lavere
- ✓ VLAN og andre policy-relaterte trafikkstyrings- og segregerings-mekanismer
- ✓ Tjenesteavtaler med underleverandører (SLA)

Adressering og ruting: I små nettverk er adressering og ruting enkelt: NAT eller offisielle adresser, ett eller et fåtall subnett og én eller i høyden et par forbindelser til Internettet. I større nettverk får vi en rekke utfordringer og mekanismer som påvirker hverandre og må avklares (se rammen for begrepsforklaringer):

- ✓ Alminnelig IP-adressering, subnetting
- ✓ Subnetting med variabel lengde
- ✓ Supernetting
- ✓ Privat adressering/NAT
- ✓ Bruk av ruting-protokoller
- ✓ Spredning/filtrering av ruter
- ✓ Peering (trafikkutveksling/formidling av tredjeparts trafikk)
- ✓ Policy-messige forhold

Sofistikerte adressestrukturer

Til tross for stadige forventninger om det motsatte, finnes det fortsatt betydelige adresse-reserver i Internettet – et forhold vi har diskutert ved tidligere anledninger i Mellvik-Rapporten (se for eksempel artikkelen "IPv6: Hvor blir du av" i nr. 100). Den viktigste årsaken til det vi kan kalle 'nye adresse-reserver' er introduksjonen av såkalt klasseløs ruting (CIDR, *Classless Interdomain Routing*) midt på 90-tallet. I kjølvannet av oppløsningen av de gamle adresseklassene har begrepene subnetting og supernetting fått utvidet betydning, samtidig med at vi har fått nye varianter – som 'subnetting med variabel lengde'.

Begrepene overlapper hverandre i noen grad, hvilket fremgår av følgende definisjoner:

- **Subnetting:** Inndeling/oppsplitting av et IP-nettverk i flere, mindre deler. IP-nettverk i denne sammenhengen var nett etter den opprinnelige klasseinndelingen – A, B eller C. I våre dager betyr 'IP-nett' den adresseblokken vi har fått tildelt fra vår ISP, som ikke følger denne modellen (se neste punkt).
- **Subnetting med variabel lengde:** En poengtering av at de opprinnelige klasseinndelingene av IP-adresser ikke benyttes, og at et nettverk ikke lenger er beskrevet av en adresse og en implisitt maske (f.eks. 62.0.0.0), men av kombinasjonen adresse og maske i antall bits (f.eks. 62.80.203.80/24).
- **Supernetting:** Det motsatte av subnetting – flere mindre nettverk kombineres til ett større. Den typiske situasjonen er å legge sammen to eller flere klasse C-nett, en mekanisme som har vært tilgjengelig siden tidlig på 90-tallet og støttes av de fleste leverandører av avanserte rutere. Ikke alle klasse C-nett kan kombineres, og det er et krav at adresseområdene er sammenhengende. Supernetting av ikke-sammenhengende adresseområder er riktignok teknisk mulig og støttes av enkelte leverandører, men er komplisert og driftsmessig krevende.

Oppløsningen av klassebegrepet i Internett-adresseringen har både forenklet og komplisert rutingen. På den ene siden er rutingtabellene i ISPenes sentrale rutere blitt vesentlig mindre. Prisen vi betaler er at det ikke lenger er like innlysende hva som er hva – og hvordan rutingen skal være. Meningen er at komplikasjonen skal gjemmes av sofistikert ruter-teknologi, hvilket som regel – men ikke alltid – er tilfelle.

Sikkerhet og beskyttelse:

Omsider er prioriteringen av sikkerhet kommet på et nivå som gjør en sikkerhetsarkitektur til en selvsagt del av nettverksarkitekturen. Her skal vi vurdere og ta hensyn til følgende områder:

- ✓ Trusselanalyser og risikovurderinger
- ✓ Regler, policy og rutiner

- ✓ Fysisk sikkerhet, bevissthet
- ✓ Sikring av datatransport (kryptering)
- ✓ Applikasjons-sikkerhet, sikring av tjenester
- ✓ Autentisering, digitale signaturer
- ✓ Sikring av inn- og ut-passeringspunkter i nettverket
- ✓ Ekstern aksess (hjemmebrukere, mobile brukere, besøkende brukere)
- ✓ Sikring av trådløse nettverk

VPN – Virtual Private Network
 PKI – Public Key Infrastructure
 NAT – Network Address Translation
 SNMP – Simple Network Management Protocol

Mekanismer og tiltak som hører med til vurderingen, omfatter blant annet følgende:

- ✓ Aksesskontroll
- ✓ Brannmurer og demilitariserte soner
- ✓ Kryptering, VPN, sikker epost
- ✓ PKI – Public Key Infrastructure
- ✓ NAT – Network Address Translation
- ✓ Pakkefiltrering
- ✓ SNMP-sikkerhet, sikring av driftsverktøy

Her dukker det opp en rekke vekselvirkninger som det er kritisk å vurdere – for eksempel: Hvordan påvirker pakkefiltrering ytelsen til utstyret som skal utføre oppgaven? Hvordan samspiller brannmur, VPN-løsning og krypterings-løsning? Tåler brannmuren VPN-belastningen og hvilke konsekvenser får kryptering av brukertrafikken for virus- og SPAM-kontroll?

Drift og styring: På samme måte som for sikkerhet, er styring og drift i dag selvsagte komponenter i arkitekturen – etter å ha vært 'attpåklatter' i altfor mange år. Arkitekturen skal ta hensyn til blant annet følgende mekanismer og forhold i denne komponenten:

- ✓ Overvåking
- ✓ Instrumentering
- ✓ Konfigurasjon og konfigurasjonskontroll
- ✓ FCAPS-komponenter⁵
- ✓ Skalering av overvåkingstrafikk
- ✓ Sentralisert kontra distribuert styring
- ✓ Integrasjon med systemadministrasjon
- ✓ Innsamling, analyse, lagring og rapportering av overvåkingsinformasjon
- ✓ ... og så videre

5 FCAPS er en relativt fersk forkortelse som har sneket seg inn på fagområdet nettverk, og sammenfatter en rekke enkeltområder som skal være dekket av en profesjonell styringsplattform: FAULT MANAGEMENT, CONFIGURATION MANAGEMENT, ACCOUNTING MANAGEMENT, PERFORMANCE MANAGEMENT, SECURITY MANAGEMENT – til sammen FCAPS.

Arkitektur-modeller

Arkitekturen skal gi oversikt, skape orden der hvor det i mange tilfeller er kaos. Vår gjennomgang i denne artikkelserien skal på sin side være idéskapende, ved siden av å gi verktøy, maler og eksempler på hvordan vi kan gå frem for å nå gode resultater uten å spille tid eller andre ressurser. I den forbindelse har vi allerede sett at modeller er viktige. Vi har blant annet diskutert flytmodeller som hjelpemiddel for å avdekke og kvantifisere datastrømmer i nettverket.

Tilsvarende tankegang brukes i forbindelse med selve arkitekturen. Vi benytter modeller som hjelpemidler for å komme raskt frem til riktige resultater. Tre slike modeller har vist seg spesielt anvendelige i praksis:

- ✓ Topologiske modeller benyttes ofte som utgangspunkt.
- ✓ Flyt-baserte modeller utnytter dataflyt som ble avdekket under flytanalysen.
- ✓ Funksjonsbaserte modeller fokuserer på én eller flere funksjoner eller egenskaper i nettverket.

Den resulterende referansearkitekturen vil i mange tilfeller inneholde mer enn én slik modell.

Topologiske modeller

Som navnet indikerer, tar modellen utgangspunkt i nettverkets topologi – et logisk og naturlig utgangspunkt selv for små nettverk. Modellen kan være fysisk, koblet til geografi, bygninger, etasjer og så videre ('LAN/MAN/WAN-modellen'), eller den kan være logisk, knyttet til nettverksnivåer: Rygggradsnett/distribusjonsnett/aksessnettverk. I begge tilfeller representerer den et godt startpunkt for mer sofistikerte arkitektur-modeller (se figur 9).



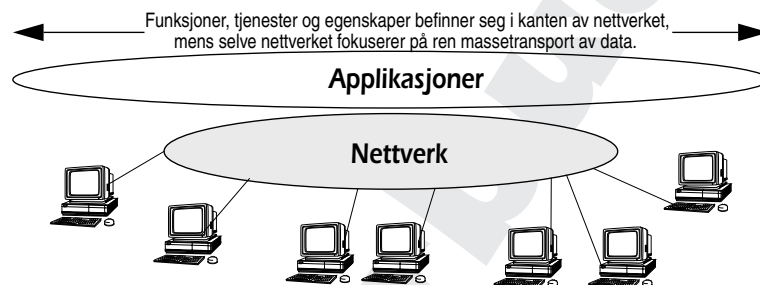
Figur 9 Topologiske arkitektur-modeller tar enten utgangspunkt i den fysiske virkelighet eller i nettverkets funksjonelle nivåer.

Flyt-baserte modeller

Flyt-baserte arkitektur-modeller følger hoved-kategoriene vi diskuterte i forbindelse med flytanalyse (flytmodeller, se forrige utgave av Mellvik-Rapporten). De mest alminnelige er *peer-to-peer* og klient/tjener, der førstnevnte utmerker seg ved sin ustrukturerte, noen vil si kaotiske natur. 90-tallets PC-nettverk var som regel av denne typen, som fortsatt beskriver en rekke anvendelser (fildelingstjenester i Internettet å la Napster hører for eksempel hjemme her).

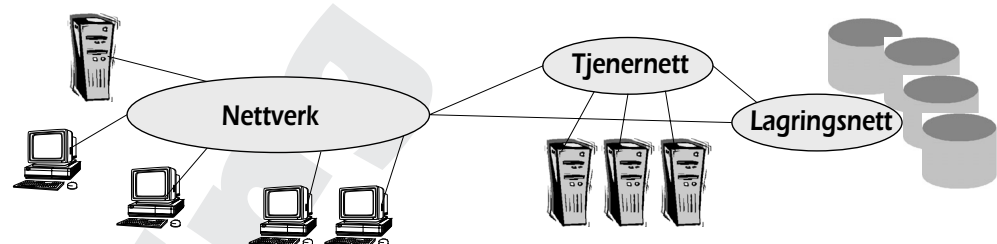
Noen karakteristika for *peer-to-peer* modellen er:

- ✓ Ingen åpenbar plassering for typiske arkitektur-elementer som tjenere og ryggradsnettverk.
- ✓ Funksjoner, tjenester og egenskaper finnes i kanten av nettverket, hvor også klientene (brukerne) finnes.
- ✓ Alle flyt er ende-til-ende.



Figur 10 En PEER-TO-PEER arkitekturmodell utmerker seg ved nettverkets 'tomhet': Alle elementer befinner seg i kanten av nettverket, som kun er en ren transportkanal, uten arkitektoniske komponenter.

Klient/tjener-modellen kan på sin side være enten todelt eller hierarkisk, avhengig av nettverkets størrelse og applikasjonenes beskaffenhet. I likhet med den tilsvarende flyt-modellen, med sine tydelige datakilder og -mottakere, gjenkjenner vi arkitektur-komponenter som sentraliserte tjenere, ryggradsnett, fordelingsnettverk og så videre.



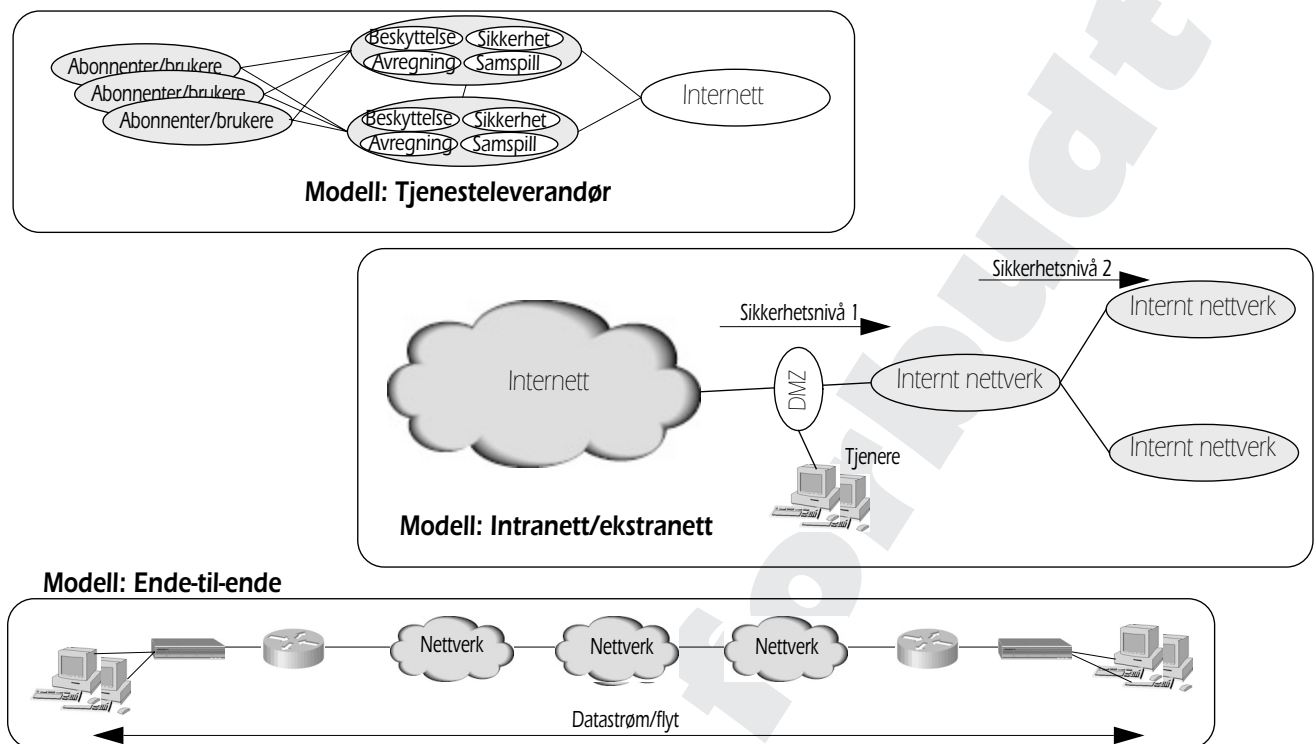
Figur 11 De fleste nettverk inneholder klient/tjener-elementer. Deres praktiske betydning og trafikkmessige egenvekt avgjør om bruk av denne arkitektur-modellen er et riktig trinn på veien mot en referansearkitektur.

Funksjonsbaserte modeller

Poenget med funksjonsbaserte modeller er å fokusere på hovedfunksjonene eller formålet med nettverket. Hva er målsettingen med totalsystemet? Dette kan i utgangspunktet høres ut som et umulig spørsmål. Mange herrer skal tjenes, utallige oppgaver skal løses, har vi egentlig en hovedoppgave? I enkelte tilfeller er svaret nei, men i de fleste ja – selv om det kanskje ikke er åpenbart.

Nedenfor gjennomgår vi kort noen eksempler som demonstrerer tankegangen, og illustrerer at det viktigste ikke er å finne en mal som passer, men å bestemme seg for hva som er viktigst – å prioritere: Levering av tjenester, levering av pålitelig og sikker konnektivitet, fokus på ytelse, på applikasjoner og brukere – eller andre varianter, se figur 12.

Tjenesteleverandør-modell: Fokus på levering av infrastruktur-tjenester, sikkerhet, avregning og tilhørende tjenester. En typisk ISP-modell som nå finner veien inn i både store og mellomstore organisa-



Figur 12 Eksempler på funksjonsbaserte arkitekturmodeller: Vi ser at det samme fysiske nettverket kan representeres på et utall ulike måter avhengig av hvordan vi prioriterer og hvilke sider vi ønsker å fokusere på.

sjoner, og gjerne passer med virkeligheten både i full skala (hele organisasjonen) og på avgrensede deler av nettverket (avdelinger, bygninger, etasjer etc.).

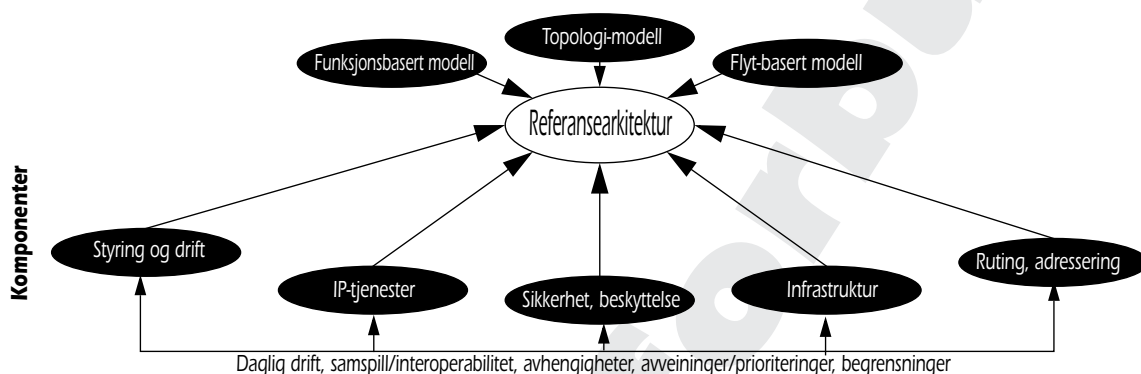
Intranett/ekstranett-modell: Fokus på sikkerhet, trafikk-segregering ('innenfor' kontra 'utenfor') og aksesskontroll. Modellen har gjerne flere sikkerhetsnivåer, og legger vekt på elementer som eksterne brukere, VPN-forbindelser og sikre aksessmekanismer.

Ytelsesmodell: Fokus på levering av ytelse etter en forhåndsdefinert profil, typisk enten 'høy' eller 'generell'. I første tilfelle er det et fåtall (gjern 5-15%) applikasjoner, utstyrsenheter og/eller brukere med spesielt høye ytelseskrav som driver arkitekturen. I det andre tilfellet er det de gjennomsnittlige kravene som er dominante (typisk 80% eller mer av totalen). De resulterende kravene til arkitekturen blir åpenbart vesensforskjellige i de to tilfellene.

Ende-til-ende modell: I små og ustrukturerte nettverk er dette som regel den riktige modellen – om ikke nødvendigvis den ønskelige. Å 'drive' en forandring over til en annen modell kan imidlertid vanskelig skje fra nettverkssiden. Dersom en slik forandring ligger i kortene, er det dog viktig at nettverksarkitekturen (og modellen) tar hensyn til dette.

Fra komponenter til helhet

Som vi har påpekt tidligere, skal arkitekturen beskrive sammenhenger mellom bestanddeler – komponenter, som vi har kalt dem. Modellene vi har gjennomgått, er verktøy på veien mot målet – referansearkitekturen – og som rent konkret består av både skisser og beskrivende dokumenter. Her samles alle elementene – sammenhenger, prioriteringer, karakteristikker, observasjoner og så videre – som illustrert i figur 13.



Figur 13 Referansearkitekturen blir oppslagsverket for designprosessen. Den skal inneholde all relevant informasjon om hva vi ønsker å oppnå, og forhold som påvirker veien til målet.

Viktigheten av å få frem sammenhenger og avhengigheter mellom komponentene i arkitekturen kan ikke overdrives. Dette kan gjøres på mange måter, og for oversiktens del er det nyttig med en tabelloppstilling omtrent som vist i figur 14.

	IP-tjenester	Styring og drift	Adressering og ruting	Sikkerhet	Infrastruktur
IP-tjenester					
Styring og drift					
Adressering og ruting					
Sikkerhet					
Infrastruktur					

Figur 14 Avhengigheter kan sammenstilles på mange måter. Tabell-form er blant de mest oversiktlige og dermed nyttige.

Neste utgave

I neste utgave begynner vi en mer detaljert gjennomgang av de viktigste enkeltkomponentene i arkitekturen, og starter med IP-tjenester. ■