

“Naming” og “trading”



INF5040

Foreleser: Olav Lysne

To design spørsmål

⌘ Navngiving

- ☒ ressursdeling krever globale lokasjonsuavhengige navn på ressurser og objekter
- ☒ hvordan konstruere navngivings skjema som skalerer i tilstrekkelig grad og som tillater effektiv avbildning fra navn til lokasjon?

⌘ Trading

- ☒ tjenester har behov for kunder
- ☒ hvordan publisere tjenester og hvordan kan klienter finne relevante tjenester (som de kanskje ikke vet navnet på)?
- ☒ Ett svar: **GULE SIDER**

Navngiving

- ⌘ En prosess som krever adgang til et objekt eller en ressurs som den ikke forvalter, må eie et navn eller en identifikator til objektet/ressursen
- ⌘ Objektets/Ressursens navn er grunnlaget for å *lokalisere* objektet
- ⌘ Lokalisering: avbilde navnet til en *kommunikasjons-identifikator* (f.eks. IP adresse/port, eller ObjektRef.)
- ⌘ Design valg for navn i DS:
 - ☒ navnerom (endelig/uendelig, flatt/hierarkisk)
 - ☒ metode for lokalisering av ressurs fra dens navn

Lokalisere ressurs fra navn



- ⌘ Lokasjon kodet i navnet

- ⌘ Distribuert navnetjener

- ⌘ Cache/broadcast

- ⌘ Forover pekere

- ⌘ Kombinasjoner av metodene over

- ⌘ Internet Domain Naming Service

- ⌘ distribuert navnetjener (inkl. replikering)

- ⌘ caching

Noen begreper

⌘ Binding

- ☑ assosiasjon mellom et navn og et objekt
- ☑ tjenste-spesifikke navn bindes til et objekt eller ressurs
- ☑ tjenste-uavhengige navn bindes til attributter om det navngitte objekt

⌘ Navneoppklaring

- ☑ finne attributtene assosiert med et navn (`look_up`)

Noen begreper II

⌘ Attributt-verdi:

- ⊞ *primitiv* (f.eks. Ethernet adresse)

- ⊞ *navn* (som igjen er bundet til et objekt eller en ressurs)

⌘ Bruker-attributter

- ⊞ e-post adresse, login navn, passord, hjemmekatalog, ...

⌘ Datamaskin-attributter

- ⊞ fysisk adresse, arkitektur, type operativsystem, ...

⌘ Tjeneste-attributter

- ⊞ versjon, aksess-protokoll, ...

Navnetjeneste

⌘ Database av bindinger

- ⊞ mellom tekstlige navn og objekt-attributter

- ⊞ `look_up : navn → attributter`

Krav til global navnetjeneste

⌘ Skalerbarhet

- ☑ effektiv handtering av vilkårlig store mengder navn
- ☑ betjene vilkårlig antall administrative domener

⌘ Lang levetid

- ☑ => må kunne absorbere endringer i navnerommets organisering

⌘ Høy tilgjengelighet

- ☑ de fleste andre systemer avhenger av navnetjenesten

⌘ Feilisolering

- ☑ lokale feil må ikke medføre total feiling av navnetjenesten

⌘ Toleranse for mistro

- ☑ ikke gjøre systemet avhengig av en eneste komponent som alle klienter må stole på

DNS

Domain Name Service

⌘ Navnetjenesten på Internettet

⌘ Brukes i all hovedsak til

☑ å oversette maskinnavn (smtp.ifi.uio.no) til IP-adresser (128.64.28.01).

☑ Fra en mailadresse å finne en maskin som kan håndtere mail for det gitte domenet.

⌘ Kan også

☑ oversette IP-adresser til maskinnavn

☑ fra et maskinnavn finne maskinarkitektur of OS-type

☑ "Velkjente tjenester" - hvilke tjenester tilbys av en maskin (f. eks telnet, FTP)

DNS Navnerom

⌘ hierarkisk (jfr. krav om skalerbarhet)

⌘ navnesyntaks:

<navnekomponent> ('.'<navnekomponent>)*

⊞ ifi.uio.no

⊞ www.yahoo.com

⌘ Domener

⊞ avsluttende del av et navn

⊞ uio.no

⊞ .com

⊞ identifiserer et *domene* -- en samling av navn med samme suffix

DNS Navnerom II

⌘ Alias

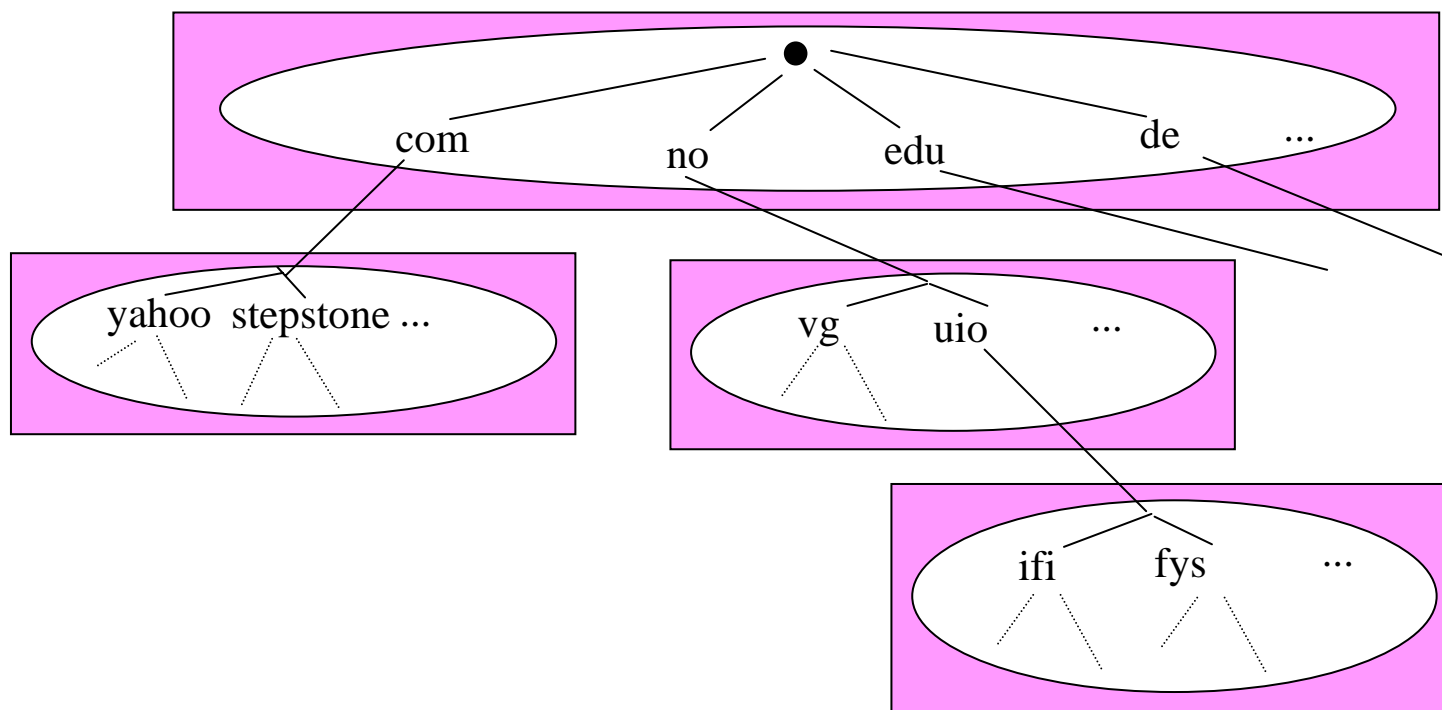
- ☒ Samme maskin kan ha flere navn
- ☒ Dette er praktisk for lokasjonstransparens (jfr. `www.ifi.uio.no`).

⌘ Hierarkisk oppdelt

- ☒ navnene deles mellom høyere-nivå domener. Disse motsvarer gjerne (men ikke nødvendigvis) forskjellige geografiske områder (`no`), samfunnsområder (`com`, `gov`) eller institusjoner og bedrifter (`vg.no`, `ifi.uio.no`).
- ☒ hvert domene er assosiert med et suffix
 - ☒ `uio.no`
 - ☒ `purdue.edu`
- ☒ administratorer kan fritt binde navn med sine respektive assosierte suffix
 - ☒ en ifi administrator kan binde navn med suffix `ifi.uio.no`
 - ☒ en `.com` administrator kan binde navn med suffix `com`

Navnetjenerere

- ⌘ DNS partiserer navnedatabasen over flere *soner*
 - ☒ hver navnetjener lagrer data for et eller flere subtrær av navnerommet samt navnet på rotkatalogen i hvert subtre



DNS Datatyper

⌘ Abstraksjon

☑ DNS lagrer records på formen

`<navn, TTL, klasse, type, verdi>`

<i>Domene navn</i>	<i>TTL</i>	<i>klasse</i>	<i>type</i>	<i>verdi</i>
www	1D	IN	CNAME	digre
digre	1D	IN	A	128.101.57.3

Domenenavnene tolkes innenfor en gitt sone

DNS Skalerbarhet og navigering

⌘ Skalerbare domain navnetjenere

- ☒ replikering & caching av deler lokalisert til steder de benyttes ofte
 - ☒ f.eks. domener nært roten er ofte replikert (jfr. *hot-spots*)
- ☒ hver tjener lagrer domenenavn og adresser til andre tjenere som kan oppklare navn som er utenfor tjenerens domene

⌘ Navigering

- ☒ iterativ og rekursiv
 - ☒ iterativ: spør en tjener først, som deretter gir deg beskjed om hvilken tjener du bør spørre i stedet
 - ☒ rekursiv: tjenerne i de forskjellige sonene kaller hverandre direkte for å oppklare navn

Hvordan oppstår "det første navn"?



⌘ Hardkodet inn

☑ "Nameserver = 129.208.64.1"

⌘ DHCP server som svarer på anrop.

Begrensinger

- ⌘ Begrensninger ved navngiving: klienten må alltid identifisere tjeneren ved navn
- ⌘ Passer ikke hvis klienten kun ønsker å bruke en tjeneste med en viss kvalitet men vet ikke fra hvem
 - ☑ Video on demand
 - ☒ lokalisere tjeneren som har tittelen og til best pris
 - ☑ Elektronisk handel
 - ☒ aksjehandel: lokalisere børs med beste tilbud ...
- ⌘ ⇒ Situasjoner der tjeneren ikke kan/bør lokaliseres basert på navn men snarere basert på kvalitative egenskaper

Trading



- ⌘ Lokalisere tjenester ut fra deres egenskaper
 - ☑ interface type
 - ☑ andre attributter (tjenestekvalitet m.m.)

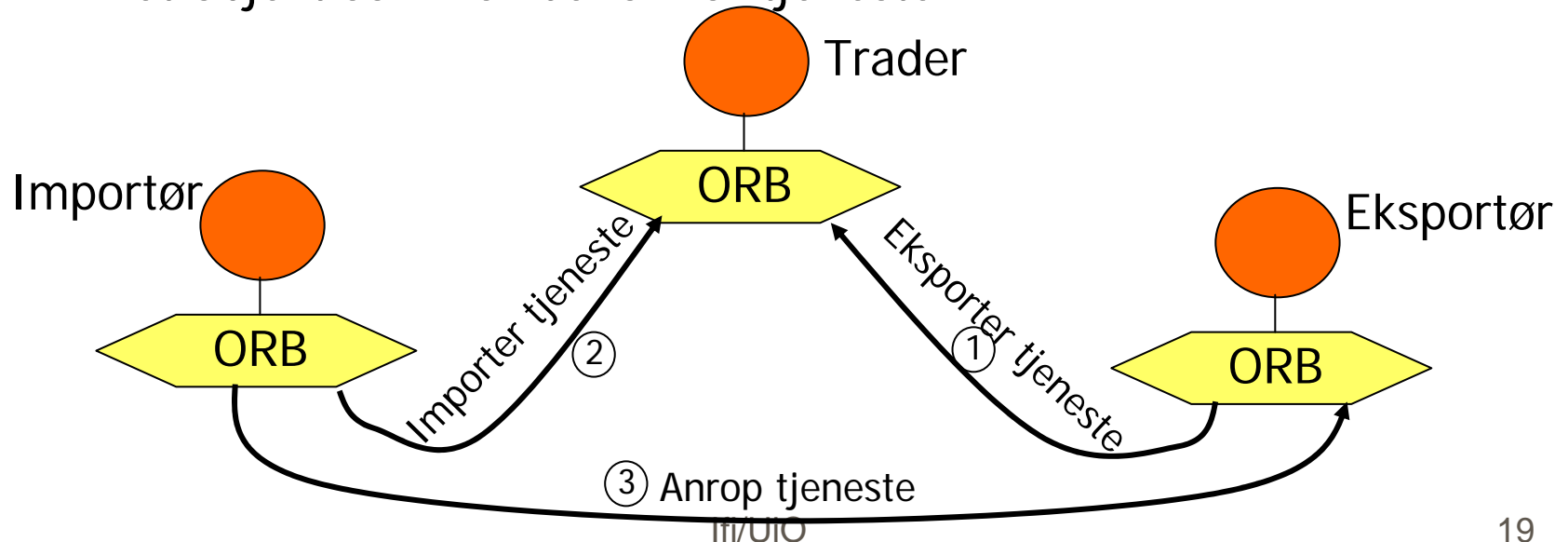
Motivasjon



- ⌘ Lokalisere objekter på en lokasjonstrasparent måte
- ⌘ Navngiving er enkel med passer ikke når
 - ☒ klienter ikke kjenner tjeneren
 - ☒ det er flere tjenere å velge mellom
- ⌘ Trading understøtter lokalisering av tjenere basert på tjener funksjonalitet og kvalitet
- ⌘ Navngiving ↔ hvite sider
- ⌘ Trading ↔ gule sider

ODP/CORBA Trader

- ⌘ En objekt-tjeneste (database) som dynamisk kan koble tjeneste-behov med tjeneste-tilbud (sein binding)
- ⌘ Eksportør
 - ☒ et objekt som tilbyr tjenester
- ⌘ Importør
 - ☒ et objekt som har behov for tjenester



Trading karakteristika

- ⌘ Felles spåk mellom klient og tjener
 - ☒ tjeneste type
 - ☒ tjenestekvalitet (QoS)
- ⌘ Tjeneren registrerer tjenester hos traderen
- ⌘ Tjeneren definerer tjenestekvalitet med garantier:
 - ☒ statisk QoS definisjon
 - ☒ dynamisk QoS definisjon
- ⌘ Klienten spør traderen om
 - ☒ en tjeneste av en bestemt type
 - ☒ på et visst kvalitetsnivå
- ⌘ Traderen understøtter
 - ☒ "service matching"
 - ☒ "service shopping"

Definisjon av tjenestetype

⌘ Tjenestetype

- ☒ funksjonalitet som tilbys av en tjeneste og
- ☒ tjenestekvaliteten som tilbys

⌘ Funksjonalitet defineres ved objekt type (grensnitt type)

⌘ Tjenestekvalitet er basert på attributterer ("properties"):

- ☒ egenskapsnavn
- ☒ egensskapstype
- ☒ egenskapsverdi(er)
 - ☒ statisk tjeneste-egenskap (f.eks. skriverhastighet)
 - ☒ dynamisk tjeneste-egenskap (f.eks. kø-lengde til skriver)
- ☒ egenskapsmodus ([obligatorisk/valgfri](#), [immutabel/mutabel](#))

Ekspert av tjenestetilbud

- ⌘ navn på tjenestetype
 - ⊗ identifiserer en tjenestetype (interfacetype & egenskapstyper)
- ⌘ objekt/interface referanse
 - ⊗ referer tjenestetilbudet
- ⌘ mengde tilbudte tjeneste-egenskaper

Import av tjenestetilbud

- ⌘ Importør spesifiserer tjenestebehov ved:
 - ⌘ Tjenesteadferd
 - ⊞ navn på tjenestetype
 - ⌘ begrensende egenskapsuttrykk (**property matching constraint**)
 - ⊞ boolsk uttrykk i et "constraint" språk
 - ⊞ evalueres over tjenesteegenskaper
 - ⊞ `PrinterType = "laser" AND Cost_per_Page < NOK10.00`
- ⌘ Utvelgelseskriterium
 - ⊞ utvalgskriterium når flere tjenestetilbud oppfyller kravene
 - ⊞ `minimize (Cost_per_Page)`

Andre eksempler på navn/trading tjenester

⌘ UDDI Universal Directory and Discovery Service (Både naming og trading)

☑ Brukes gjerne sammen med Web-Services

⌘ CORBA Naming Service

⌘ ODP/CORBA Trader