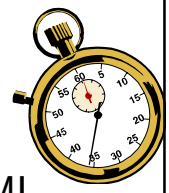
Programvarekomponenter og distrbuerte system

INF 5040 høst 2005

foreleser: Frank Eliassen

En historie om mellomvare

- Første generasjons mellomvare
  - Utelukkende basert på *klient-tjerner modellen*
  - Eksempler inkluderer Open Group's DCE
- Andre generasjons mellomvare
  - Basert på *distribuert objekt-teknologi*
  - Eksempler inkluderer CORBA and Java RMI
- Tredje generasjons mellomvare?
  - Basert på kommende *komponent-teknologi*

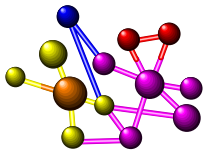


Fremveksten av komponent-teknologier

- Hva er en komponent [Szyperski]?

  " a unit of *composition* with *contractually specified interfaces* and explicit *context dependencies* only"

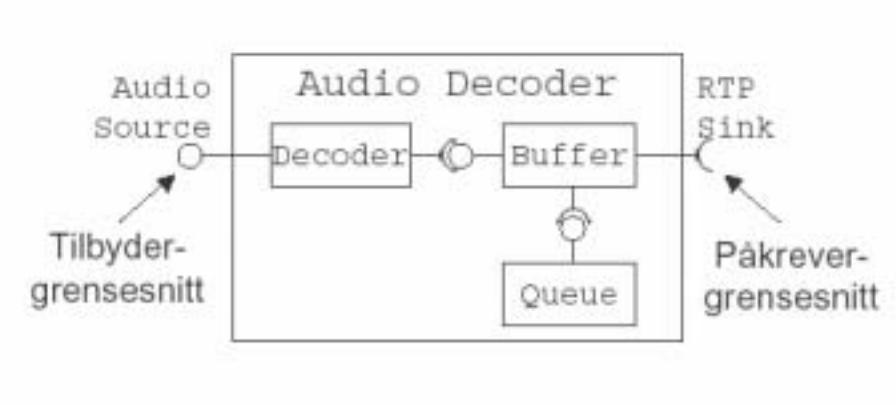
  "in this context, a component can be *deployed independently* and is subject to *third-party composition*"



Rasjonale for komponenter

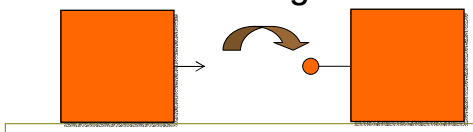
- Tid til markedet
  - Forbedret produktivitet/ redusert kompleksitet
  - Fokus på gjenbruk
- Programmering ved montasje (fabrikasjon) i stedet for utvikling (engineering)
  - Reduserte krav til kunnskaper
- Viktigste fordel: utvikling av tjener-siden?
  - (Se EJB og CORBA Component Model seinere)

Gjenbruk av komponenter



Komposisjon I

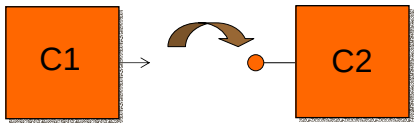
- Komponenter og komposisjon
  - Komposisjon er den grunnleggende metode for konstruksjon, utvidelse og gjenbruk i komponent-basert programvare-utvikling
  - Jfr (implementasjons) arv i objekt-orienterte tilnærminger

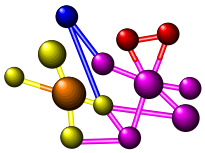


"Komponenter er til for komposisjon"

Komposisjon II

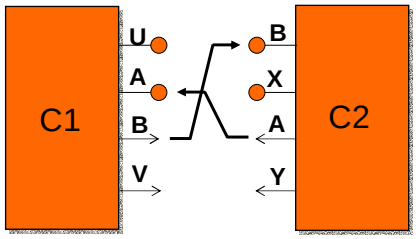
- Connection-oriented programming
  - Komposisjon av pre-fabrikerte komponenter
  - Inngående- og utgående grensesnitt
    - (provided/required interfaces)
    - Reflekterer retningen på metodekall
      - Ikke retningen til dataflyten
  - Utgående grensesnitt
    - De metodekall en komponent potensielt kan utstede

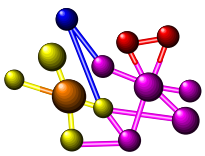




Tredjeparts-komposisjon

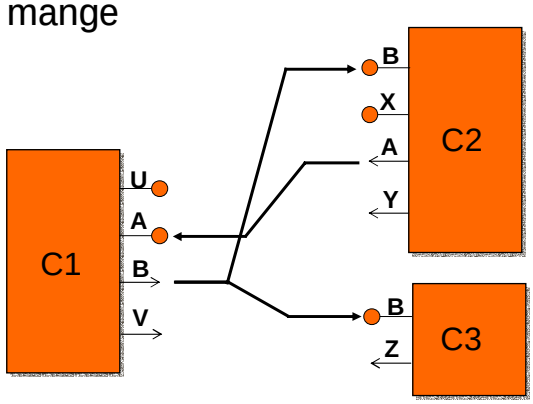
- Komposisjonen kan gjøres av tredjepart
- Eksempel
  - Connections, outgoing and ingoing interfaces
  - Forbinder “matchende” grensesnitt
  - Kan gjøres under kjøretid
    - Kan typisk gjøres ved å sette et passe attributt i komponenten med det utgående grensesnitt (setXXX)

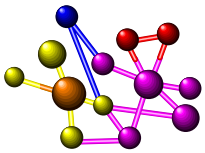




Mange "connection" mønstre

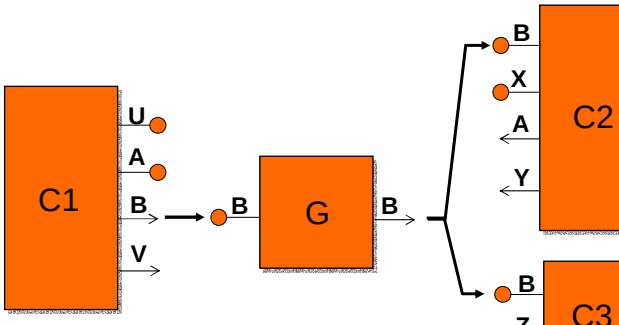
- En-til-en, en-til-mange, mange-til-en, mange-til-mange

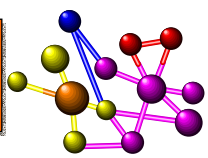




"Indirection"

- Frakobling av inngående og utgående grensesnitt ved mellomliggende komponenter





Komposisjon vs arv

- The fragile base class problem (Szyperski)
  - Med bruk av arv, er det ofte vanskelig å modifisere en superklasse (base class) uten å påvirke uavhengig utviklede subklasser
- Fordeler ved komposisjon
  - Bygger på "*message forwarding*" ikke arv
  - "Message forwarding" er enklere men mindre uttrykksfu enn arv.
- Komposisjon krever mer eksplisitt koding av relasjoner mellom entiteter (*designed in* vs *patched in*)

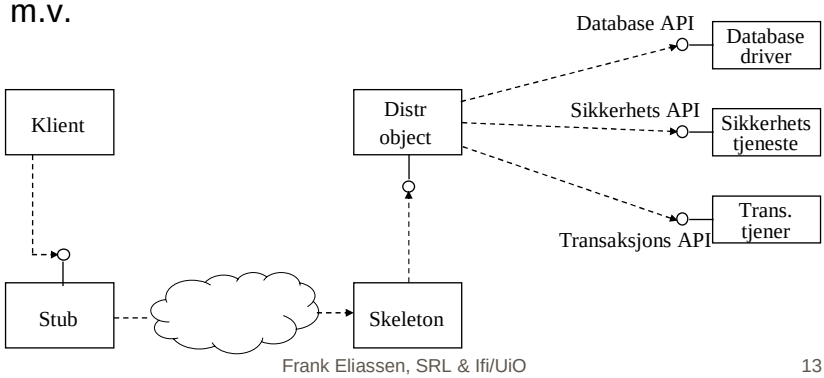
Bakgrunn for Java og CORBA komponentmodeller

- Kjente problemer med CORBA og Java-RMI
  - Hvordan utplasserer jeg min applikasjon?
  - Hvilke tjenester vil være tilgjengelige på en gitt vertsmaskin?
  - Hvem vil aktivisere mine objekter?
    - Jfr forelesning om designutfordringer
  - Hvem forvalter mine objekters livssyklus?

=> Vi trenger en standard utviklings, utplasserings og kjøretidsomgivelse for distribuerte objekter (CORBA, Java)

Eksplisitt mellomvare

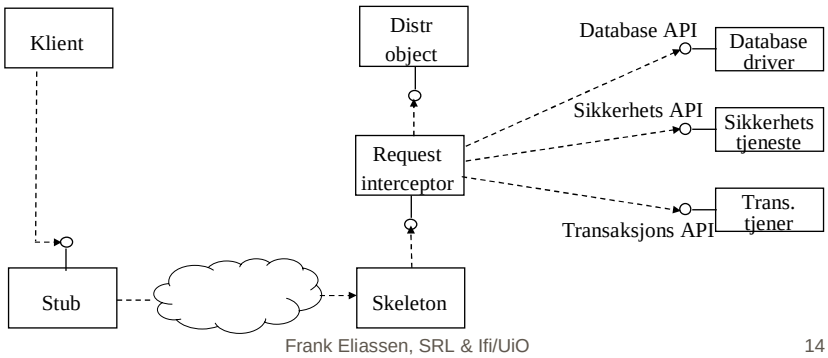
- Programmerer direkte mot en mellomvare API
- Applikasjonslogikken må sammenveves med logikk for livssyklushandtering, transaksjoner, sikkerhet, persistens, m.v.



Frank Eliassen, SRL & Ifi/UiO 13

Implisitt mellomvare

- Logikk for livssyklushandtering, transaksjoner, sikkerhet, persistens, m.v. handteres av mellomvaren
- Behov for mellomvaretjenester erklæres separat og kan seinere endres uten å endre applikasjonskoden
- Mellomvaren kan endres uten å endre applikasjonskoden



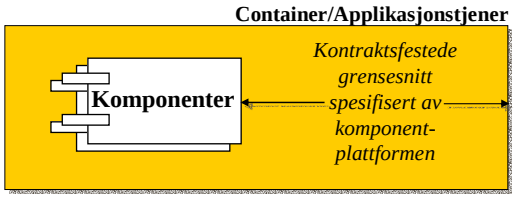
Frank Eliassen, SRL & Ifi/UiO 14

Komponentplattform

- En standard utviklings, utplasserings og kjøretidsomgivelse kan utformes som en mengde kontraktsfestede grensesnitt
- Kontrakten inngås mellom komponenter og en komponentplattform
- Komponentplattformen definerer reglene for utplassering (installasjon), komposisjon og aktivering av komponenter.

En implementasjon av en komponent-plattform kalles gjerne en container

- Containerens ansvar
  - livssyklushåndtering
  - systemtjenester
  - sikkerhet
  - dynamisk installasjon og aktivering av nye komponenter



Kontrakter

- Hva inneholder en kontrakt?
  - *Mengde provided* grensesnitt.
    - Noen av disse kan være påkrevd av komponentplattformen
  - *Mengde required* grensesnitt.
    - Disse må tilbys av andre komponenter tilgjengelig i containeren
  - Pre og post betingelser/invarianter
  - Annet (ikke-funksjonelle krav)
- Funksjoner defineres både syntaktisk og semantisk
  - int add(int a, int b)
  - pre: a + b <= Integer.MAXINT
  - post: result' = a + b
- Ekstra-funksjonelle krav
  - Garanti: Retur innen 10 ms
  - Betingelse: Trenger 1000 CPU-sykler

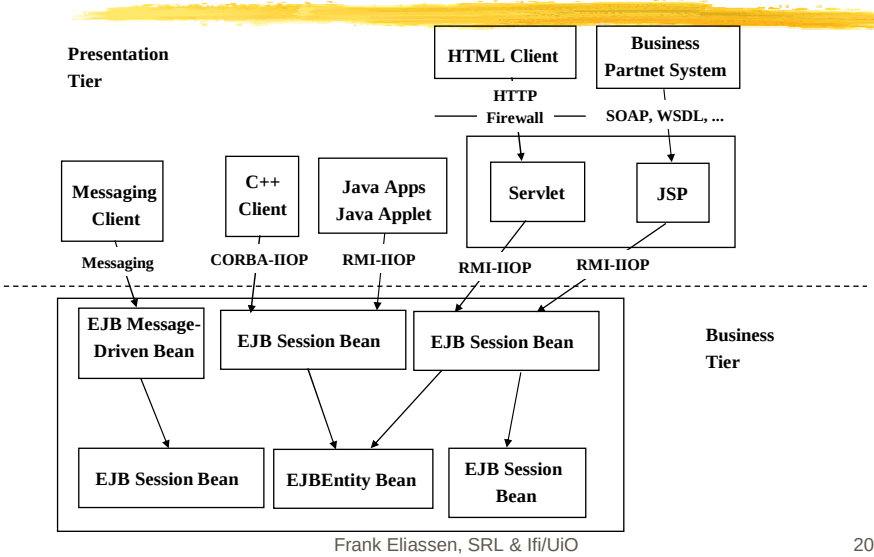
Nøkkelspillere

- OMG og komponenter
  - CORBA v3 standarden med CORBA Component Model (CCM)
- Microsoft og komponenter
  - Utvikling av COM/DCOM, COM+ og .NET
- SUN og komponenter
  - Utvikling av Java Beans og EJB

Enterprise Java Beans (EJB)

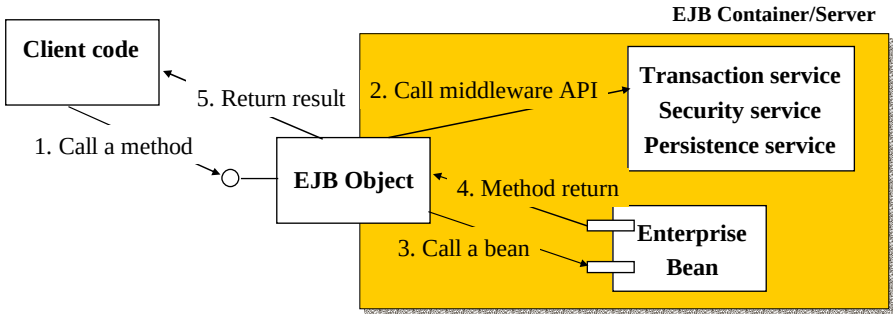
- Komponent-arkitektur for utplasserbare ("deployable") tjener-side komponenter i Java.
- Litteratur:
  - http://java.sun.com/j2ee/overview.html
  - http://www.theserverside.com/books/masteringEJB/
    Part I: gir introduksjon til komponentarkitekturer og EJB
- Tre typer enterprise beans
  - Session beans (verb)
    - Transiente, applikasjonslogikk (forretningsregler …)
  - Entity beans (substantiv)
    - Persistente, data-relatert logikk (oppdatere tilstand til entiteter)
  - Message driven beans
    - Logikk for å motta asynkrone meldinger og evt kalle session beans

Klient-interaksjon med EJB komponent system



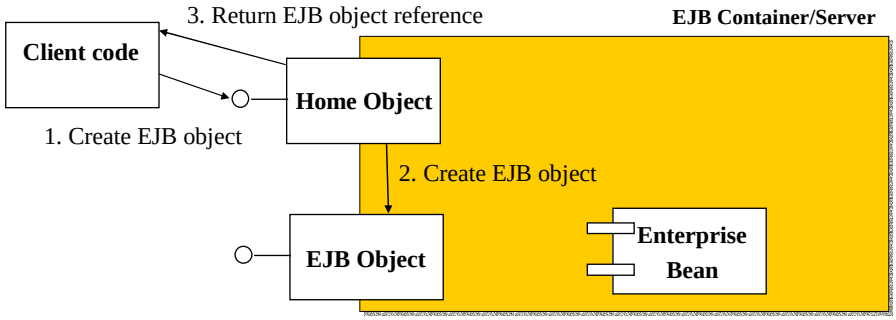
EJB objektet

- Request Interceptor
  - Delegerer forespørsler til bønner
  - Representerer implisitt mellomvare



Home objektet

- EJB objektfabrikk
  - Oppretter, lokaliserer og fjerner EJB objekter
  - Klienter lokaliserer Home objekter vha navnetjeneste



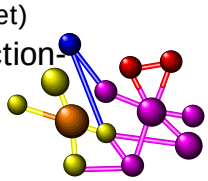
Innpakking og utplassering

- Innpakking
  - For å levere og utplassere en komponent kreves et standardisert arkivformat som pakker inn separate filer (komponent-kode og meta-data)
  - Ejb-jar fil
    - Fil som inneholder all informasjon som er nødvendig for å utplassere en komponent i en container-omgivelse

- Deployment descriptor
  - XML fil som beskriver konfigurasjon (ulike grensesnitt og bean-klasser), krav til mellomvare-tjenester, etc

"Connection-oriented programming" og EJB

- Ingen støtte for connection-oriented programming!!
  - Følger tradisjonell objekt-orientert komposisjon (tredjepart kan ikke binde EJBer)
  - Styrken er automatisk komposisjon av komponent-instanser med passe tjenester og ressurser
    - Automatisk konfiguering av nødvendig implisitt mellomvare ut fra behov spesifisert i en deployment-descriptor (transaksjoner, persistens og sikkerhet)
  - (JavaBeans har imidlertid støtte for connection-oriented programming)



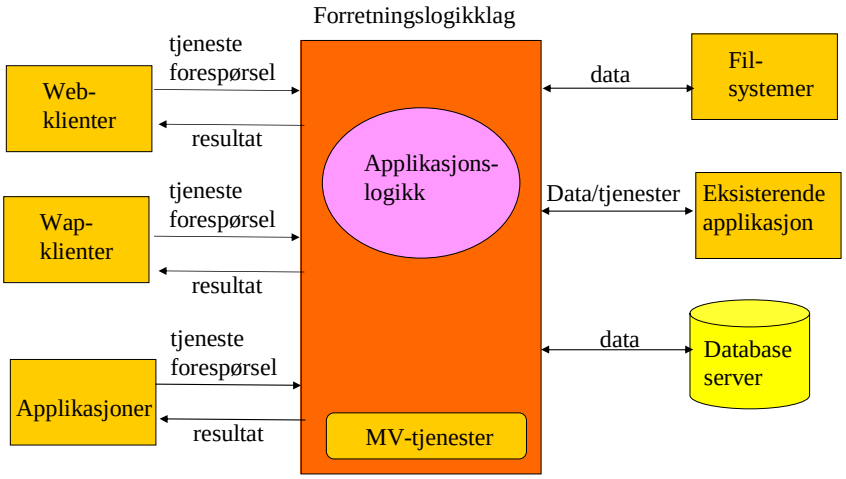
Java 2 Enterprise Edition

- En (spesifikasjon av en) plattform for utvikling og kjøring av forretningskritiske systemer
- Definerer
  - tjenester
  - bibliotek
  - protokoller
  - kjøremiljø
- Målet med J2EE er å gjøre det lettere å utvikle applikasjoner med flerlagsarkitektur
- Inkluderer EJB komponent-arkitektur

Flerlagsarkitektur

- Krav om åpenhet og distribusjon
- Større fleksibilitet enn tradisjonelle klient-tjener systemer
- Deler presentasjon, data og forretningslogikk i egne programkomponenter, uavhengig av presentasjon og datarepresentasjon
- Flerlagsarkitektur bygger på komponentmodeller

Flerlagsarkitektur



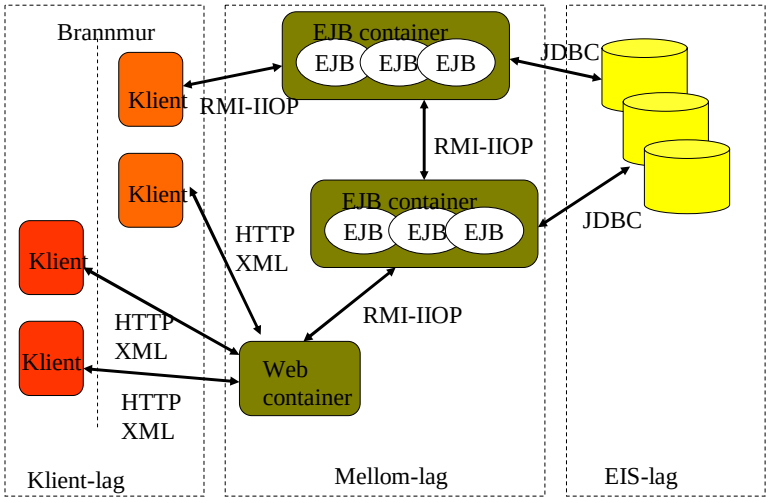
J2EE API spesifikasjoner

- **Enterprise Java Beans**: Komponentmodell for å bygge gjenbrukbare tjenerkomponenter
- **Java Database Connectivity (JDBC):** Javagrensesnitt mot relasjonsdatabaser
- **Java RMI over the Internet-ORB Protocol (RMI-IIOP):** Fjernmetodeanrop mellom Java VM basert på IIOP.
- **Java Message Service:** Asynkron kommunikasjon vha meldinger
- **Java IDL:** En Java CORBA ORB som implementerer et subsett av CORBA spesifikasjonen
- **Java Server Pages (JSP):** Dynamisk generering av web-sider
- **Java Servlets:** Servlets er komponenter som utplasseres på en webtjener
- **Java Transaction Service (JTA):** Transaksjonstjeneste
- …

EJB/J2EE vs CORBA

- EJB komplementerer CORBA
  - Mange EJB-tjenere bygger på CORBA-implementasjoner
  - EJB-teknologi gjør det lettere å bygge applikasjoner på toppen av CORBA sin infrastruktur

Java 2 Enterprise Edition

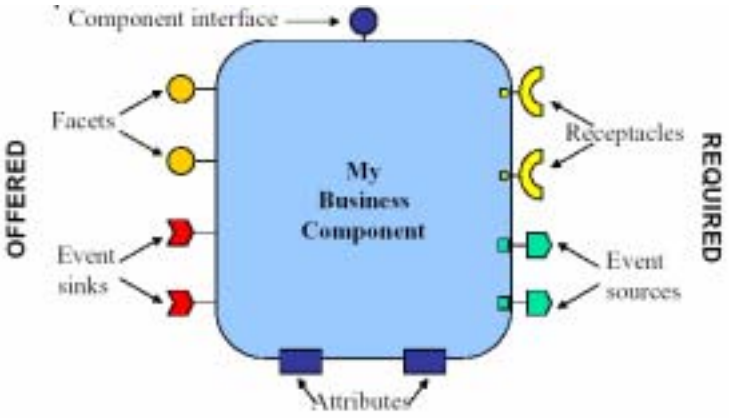


CORBA Component Model (CCM)

- Hva er CCM?
  - En *språkuavhengig*, komponentmodell for tjener-siden av fler-lagsarkitekturen som understøtter implementasjon, forvaltning, konfigurering og utplassering av CORBA applikasjoner
- Viktige egenskaper
  - En underliggende *komponentmodell*
  - En innpakkingsteknologi for utplassering av binære, *flerspråklige* eksekverbare enheter
  - Et *container rammeverk* som tilbyr implisitt mellomvare for sikkerhet, transaksjoner, persistens og hendelsesbasert kommunikasjon

En CORBA komponent

- Støtte for connection-oriented programming
  - Connect/disconnect operasjoner på Receptacles
  - Eller basert på script-språk (del av CCM deployment descriptor)



Microsoft COM ...

- Tilby en komponent objekt-modell basert på prinsippene om binær innkapsling og binær kompatibilitet
  - *Binær innkapsling*: klienter må ikke rekompileres selv om tjenerobjektet endrer seg
  - *Binær kompatibilitet*: klient- og tjenerobjekter kan utvikles med forskjellige utviklingsomgivelser og forskjellige språk
- Grunnleggende mekanisme for å oppnå binær innkapsling og binær kompatibilitet i COM:
  - skille mellom grensesnitt og implementasjon
- Støtte for spesifikasjon av utgående grensesnitt
- COM er en proprietær og de-facto standard
- COM+ adderer tjenester og interceptors til COM
- .NET introduserer CLR (Common Language Runtime) m.m.
- Mer seinere (studentpresentasjoner)

Oppsummering

- Komponenter
  - Programmering etter LEGO-prinsippet
  - Kontraktsmessige grensesnitt og komposisjon
  - Støtte for connection oriented programming
- Komponentarkitektur
  - Spesifiserer kontraktsfestede grensesnitt mellom komponenter og applikasjonstjenere.
  - Java: EJB, CORBA: CCM, Microsoft: COM+/.NET
- Java 2 Enterprise Edition (J2EE)
  - en standard utviklings-, utplasserings- og kjøretidsomgivelse for distribuerte EJB objekter