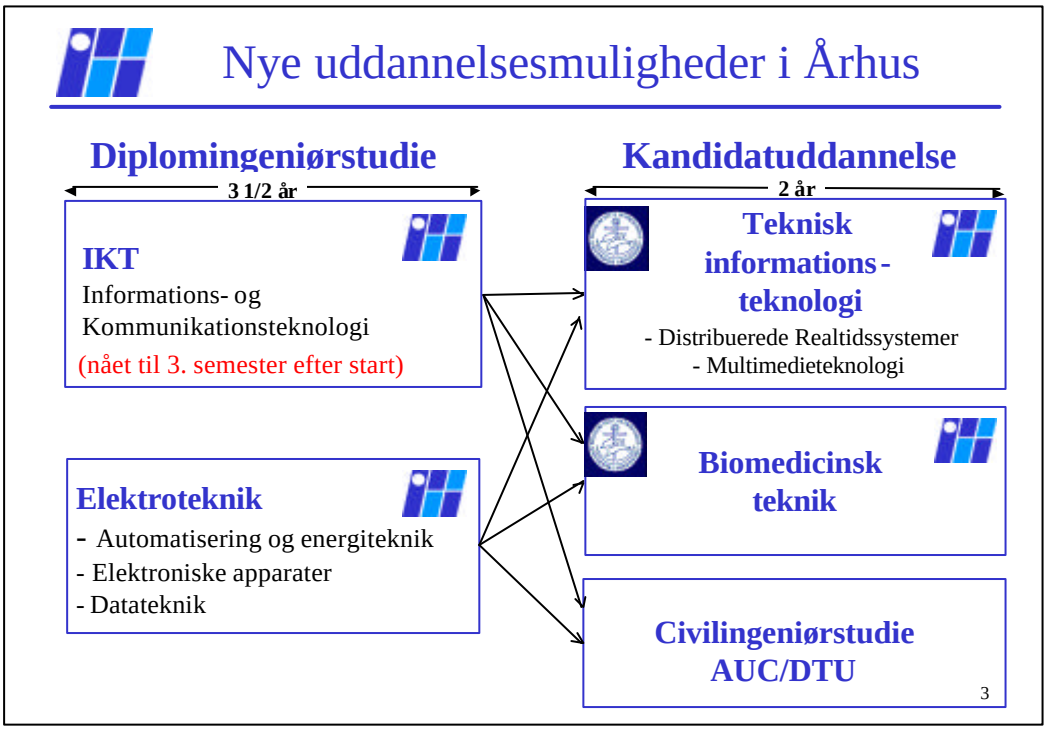
Aktuelle SW teknologier for distribuerede systemer

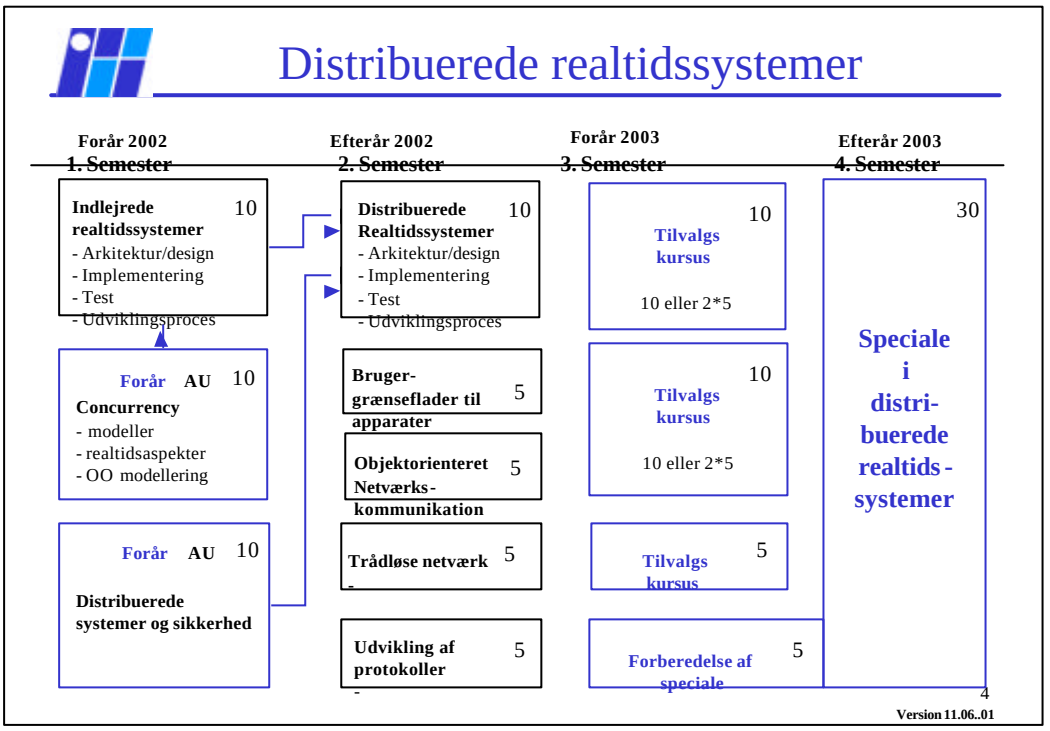
OO-UML brugergruppemøde 25. oktober 2001
Finn Overgaard Hansen
Ingeniørhøjskolen i Århus - www.iha.dk
Elektro- og IKT-afdelingen
foh@e.iha.dk

Agenda

- Introduktion til distribueret softwareteknologi
- Middelware teknologier:
  - CORBA
  - Java RMI
  - DCOM
- Auto-discovery teknologier
  - Jini, UPnP, Salutation, HAVI
- Opsummering

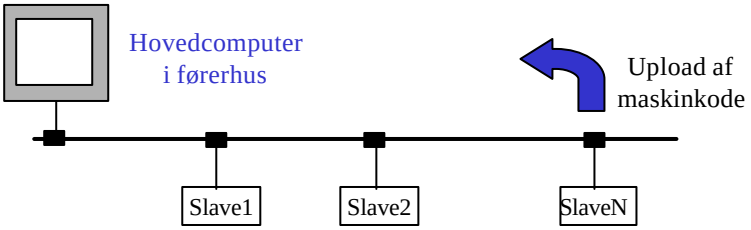
2


Nye uddannelsesmuligheder i Århus
Diplomingeniørstudie
3 1/2 år
IKT
Informations- og Kommunikationsteknologi
(nået til 3. semester efter start)
Elektroteknik
- Automatisering og energiteknik
- Elektroniske apparater
- Datateknik
Kandidatuddannelse
2 år
Teknisk informations-teknologi
- Distribuerede Realtidssystemer
- Multimedieteknologi
Biomedicinsk teknik
Civilingeniørstudie AUC/DTU
3


Distribuerede realtidssystemer
Forår 2002
1. Semester
Efterår 2002
2. Semester
Forår 2003
3. Semester
Efterår 2003
4. Semester
Indlejrede realtidssystemer 10
- Arkitektur/design
- Implementering
- Test
- Udviklingsproces
Forår AU 10
Concurrency
- modeller
- realtidsaspekter
- OO modellering
Forår AU 10
Distribuerede systemer og sikkerhed
Distribuerede Realtidssystemer 10
- Arkitektur/design
- Implementering
- Test
- Udviklingsproces
Bruger-grænseflader til apparater 5
Objektorienteret Netværks-kommunikation 5
Trådløse netværk 5
Udvikling af protokoller 5
10
Tilvalgs kursus
10 eller 2*5
10
Tilvalgs kursus
10 eller 2*5
Tilvalgs kursus 5
Forberedelse af speciale 5
30
Speciale i distri-buerede realtids-systemer
4
Version 11.06..01

En case på et distribueret system

- Dronningborg mejetærsker (ca. 1986/87)
  - Der kunne tilsluttes nye ”devices” til en fælles bus - uden at hovedcomputeren skulle omprogrammers



- Løsning:
  - Hver device uploadede den nødvendige maskinkode - til hovedcomputerens RAM lager med display rutiner mm.
  - Samme maskinkode i hovedcomputer som i satellit microprocesserne
  - Definering af faste memorygrænser for upload af nye programmer
  - En simpel low cost løsning

5

Hvorfor distribuering ?

- Distribuering i problemområdet
  - Vejrprognosesystem: Central server med decentrale målestationer (klienter) rundt omkring i landet
  - Pervasive computing: mobile trådløse enheder
    - Bærbare computere, mobiltelefoner, PDA’er
- Distribuering valgt som systemarkitektur
  - For at opnå fleksibilitet og større genbrug
  - For at spare ”wiring” og opnå en bedre pålidelighed
  - For at opnå nødvendig ”performance”
  - For at kunne integrere forskellige platforme og program-meringssprog

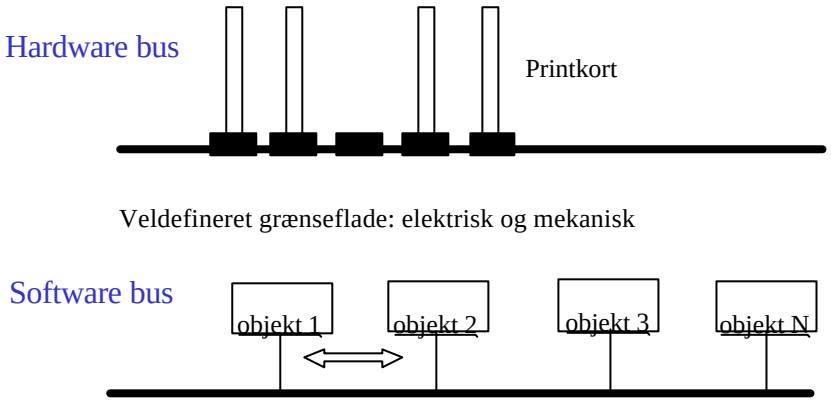
6

Hvad vi gerne vil opnå

- Software udviklere vil gerne:
  - Kunne programmere distribuerede systemer uden at skulle programmere hele kommunikationslaget
  - Kunne kalde lokale og remote objekter på samme måde
  - Kunne genbruge eksisterende kode skrevet i forskellige sprog
- Brugere og installationsfolk vil gerne:
  - kunne tilslutte nye enheder uden en omfattende konfigurering
    - nye typer af mobile ”devices” forudsætter ”plug and play” for at være anvendelige i praksis

7

Løsning: Software bussen



Programmøren ønsker at kunne skrive følgende ved kodning af objekt 1:

pObjekt2->print(file)

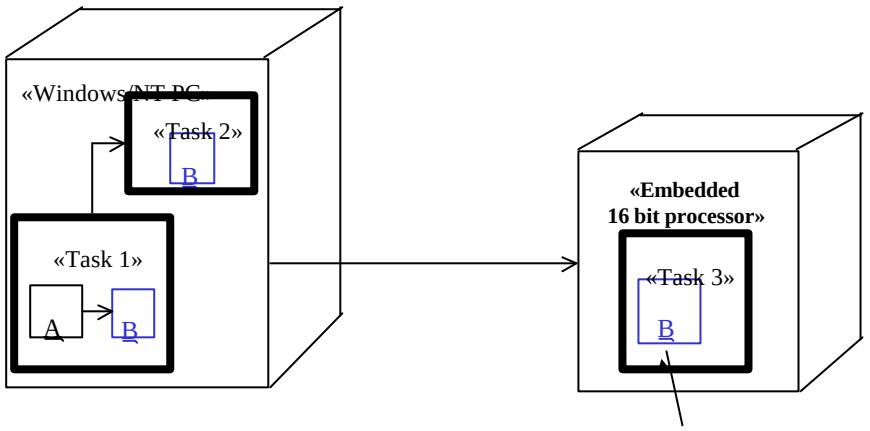
8

To former for kommunikation

- Synkron kommunikation
  - RPC - Remote Procedure Call
    - En klient kan kalde procedurer i server programmer, der kører i andre processer eller på andre computere
  - Remote Method Invocation (RMI)
    - En objektorienteret udgave af RPC
    - Objekter i forskellige processer kan kommunikere ved at kalde hinandens operationer (metoder) (RMI anvendes her som generel term)
- Asynkron kommunikation
  - Event - notification
    - Et objekt kan abonnere på at få en notification fra et andet objekt når en bestemt hændelse (event) indtræffer
    - Kaldes også for publish/subscribe

9

Objektorienteret kommunikation



Objekt A ønsker at kalde operationer i objekt B på samme måde uanset om objekt B er i:

- samme task (Task1),
- et andet Task 2
- eller befinder sig på en helt anden processor

Kan indkapsle ikke OO kode som f.eks. C som et objekt

10

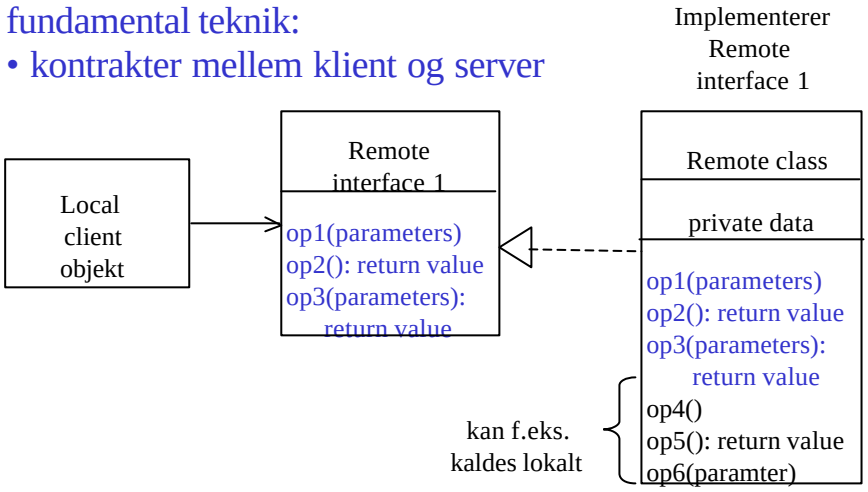
Lokations transparens

- Ved RMI:
  - Et klientobjekt der kalder et serverobjekt behøver ikke at vide om det er et lokalt objekt eller et objekt på en anden maskine
- Ved event baseret kommunikation:
  - objekter der generer hændelser og de objekter der modtager *notifications* herpå behøver ikke at kende hinandens lokation
- I praksis:
  - Man tilstræber at kaldesyntaksen er den samme
  - Klienter bør dog vide om det er et lokalt objekt eller et remote objekt f.eks. er fejlhåndteringen forskellig

11

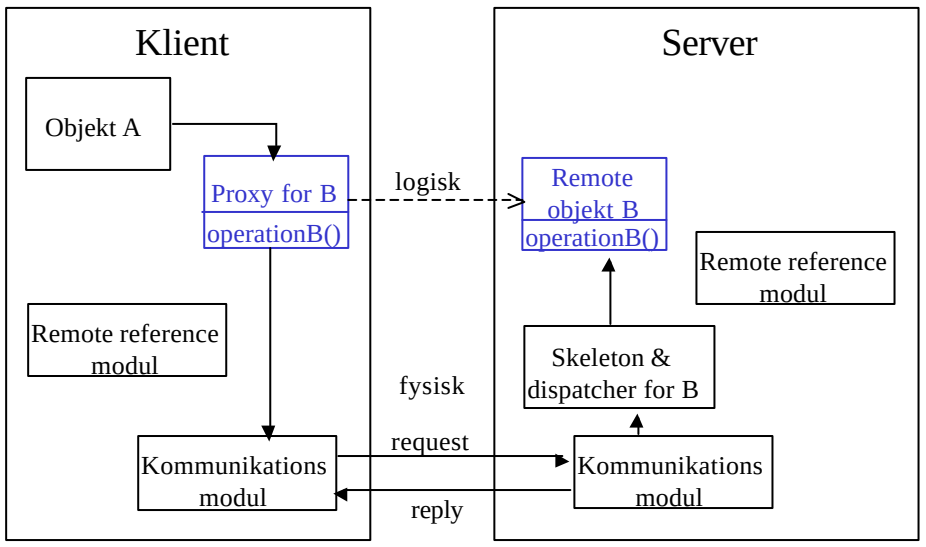
Remote interfaces

Anvendelse af interfaces er en fundamental teknik:
• kontrakter mellem klient og server



12

Struktur for *middelware* SW + proxy



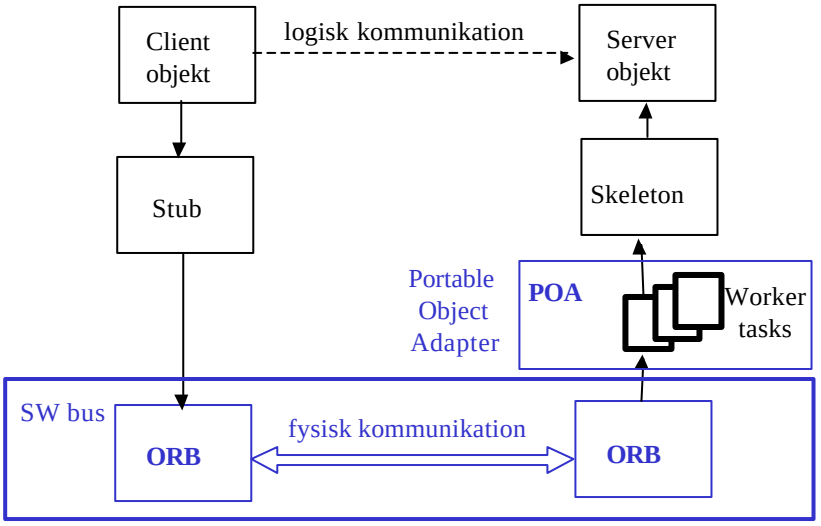
13

CORBA

- CORBA (Common Object Request Broker Architecture)
  - En OMG standard (Object Management Group) introduceret i 1991 - CORBA 2.0 i 1996 - CORBA 3.0 er på vej.
  - OMG er en sammenslutning af ca. 800 firmaer
  - CORBA er tilgængeligt på over 50 operativsystemer (f.eks. Windows9x/NT/2000, UNIX, Linux og embedded RTOS)
  - Supporterer syv programmeringssprog (C++, Java, C, Ada, COBOL, Smalltalk og Lisp)
  - Central mekanisme: ORB - Object Request Broker
  - Plus et stort antal CORBA services
    - f.eks. Naming, Event & Notification, Concurrency, Persistency
- Reference: www.omg.org

14

CORBA arkitektur

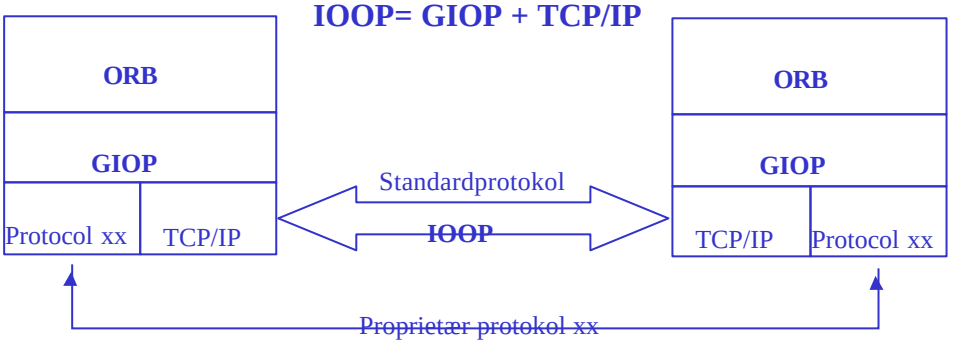


15

CORBA protokoller

GIOP: General Inter-ORB Protocol
IOOP: Internet Inter-ORB protocol

IOOP= GIOP + TCP/IP

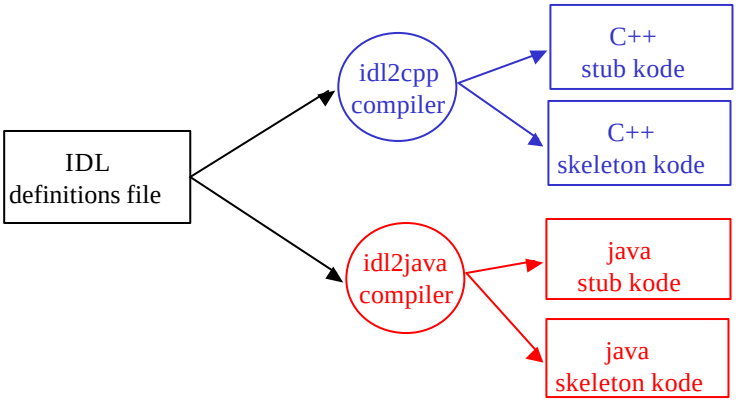


Pt. proprietære udvidelser - standard på vej for *pluggable protocols*

16

IDL: Interface Definition Language

- Alle interfaces til objekter defineres i IDL
  - IDL ligner klassedefinitioner i C++ og Java
  - IDL filen oversættes til det aktuelle sprog vha. IDL kompilere

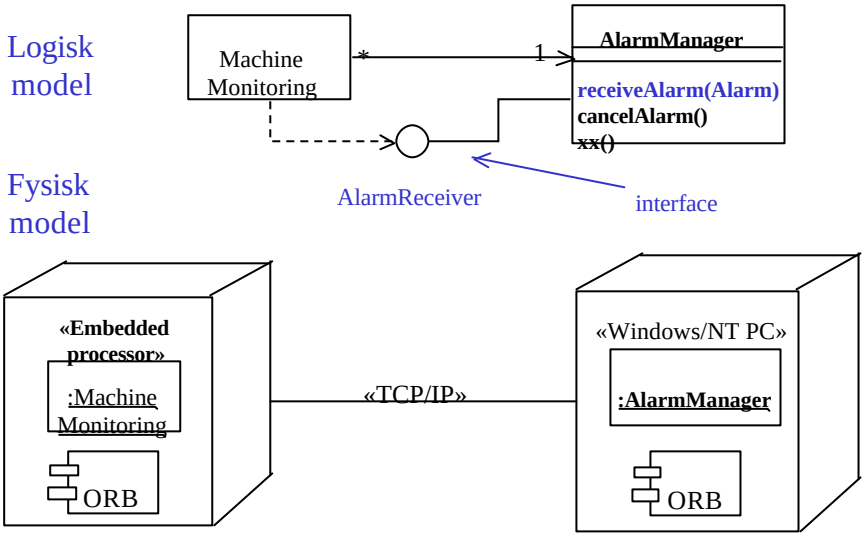


17

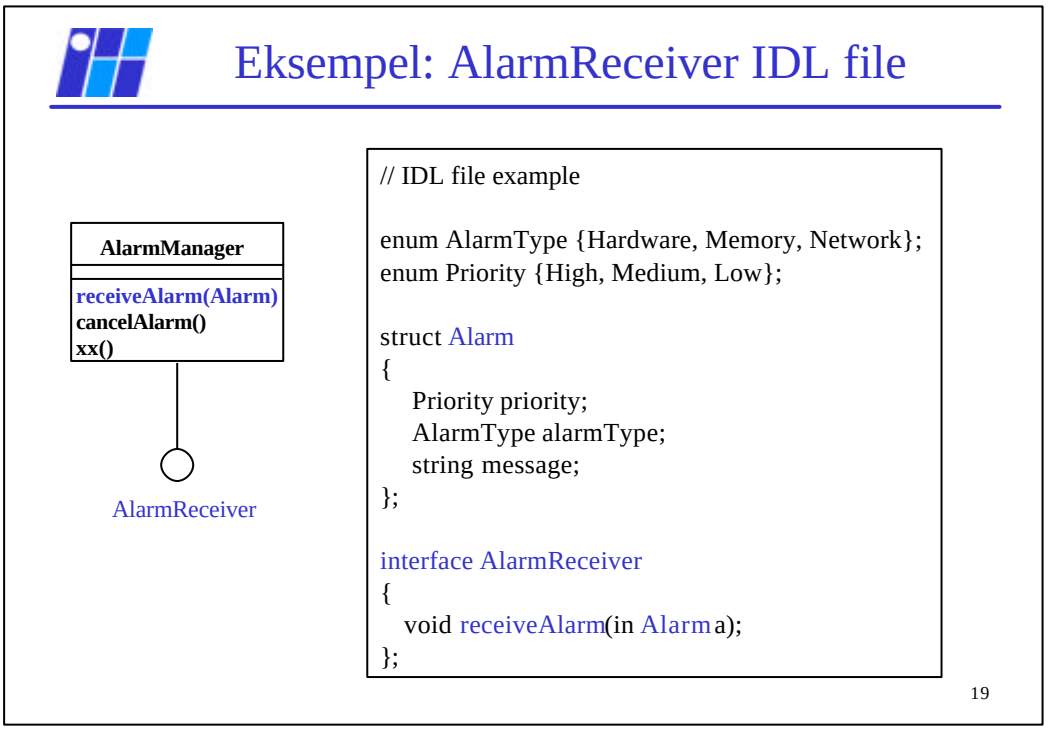
Maskinovervågnings eksempel

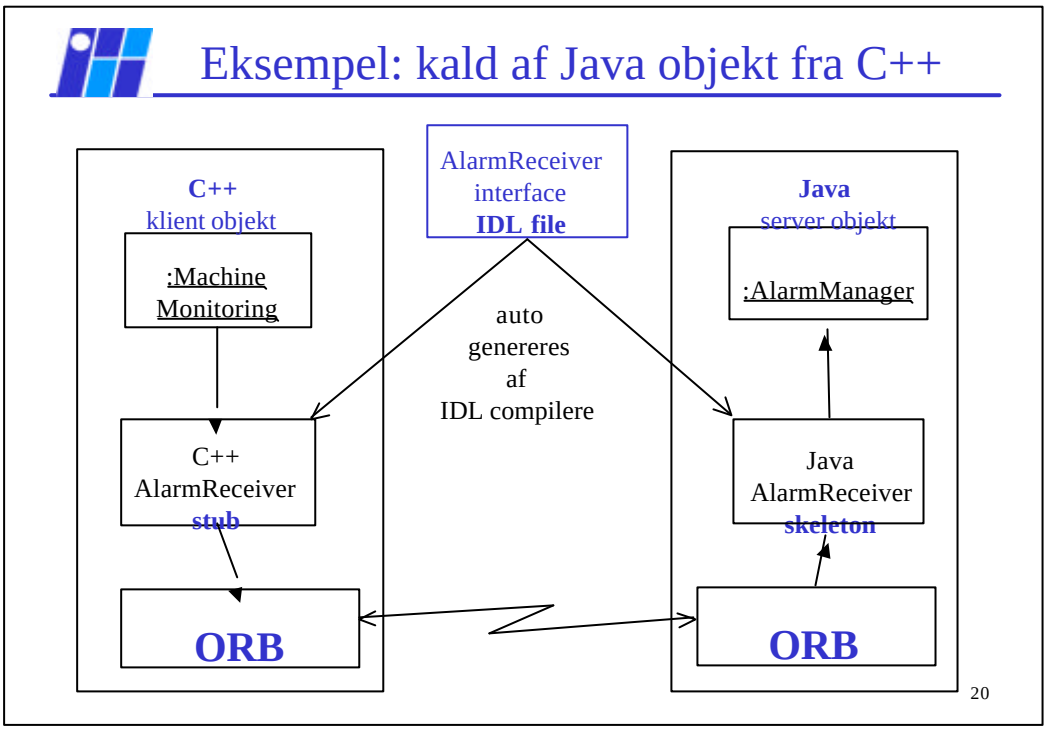
Logisk model

Fysisk model



18


Eksempel: AlarmReceiver IDL file
AlarmManager
receiveAlarm(Alarm)
cancelAlarm()
xx()
AlarmReceiver
// IDL file example
enum AlarmType {Hardware, Memory, Network};
enum Priority {High, Medium, Low};
struct Alarm
{
Priority priority;
AlarmType alarmType;
string message;
};
interface AlarmReceiver
{
void receiveAlarm(in Alarm a);
};
19


Eksempel: kald af Java objekt fra C++
AlarmReceiver
interface
IDL file
C++
klient objekt
:Machine
Monitoring
C++
AlarmReceiver
stub
ORB
auto
genereres
af
IDL compilere
Java
server objekt
:AlarmManager
Java
AlarmReceiver
skeleton
ORB
20

CORBA styrker/svagheder

- Styrker:
  - En åben standard, der er meget anvendt - og videreudvikles
  - Supporterer mange sprog pt. syv - let at tilføje flere
  - Uafhængig af operativsystem
  - Er venlig overfor Java (Java 2 er CORBA enablet)
  - Har et stort antal nyttige services
- Svagheder:
  - Kommercielle CORBA implementeringer er ekstremt dyre
  - Standard CORBA er ekstremt pladskrævende og komplekst
  - Virker ikke godt gennem firewalls
  - Dårlig opførsel i *"disconnected"* environments

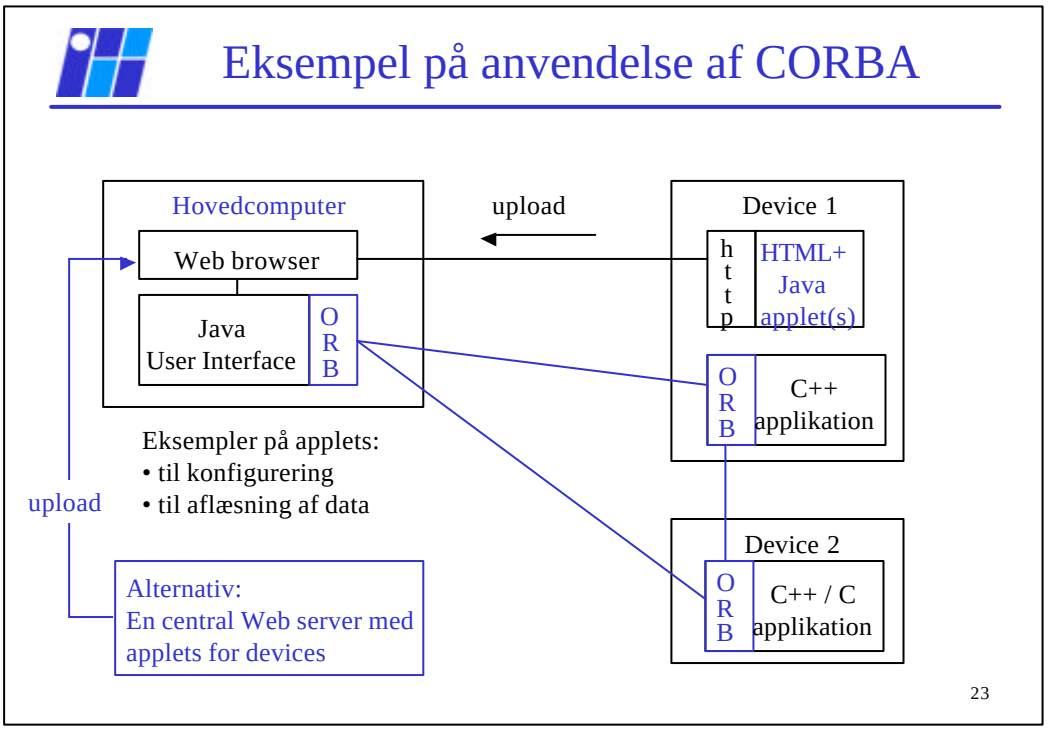
Kilde: Mike Preradovic, Intrinsic Software, ESC2001

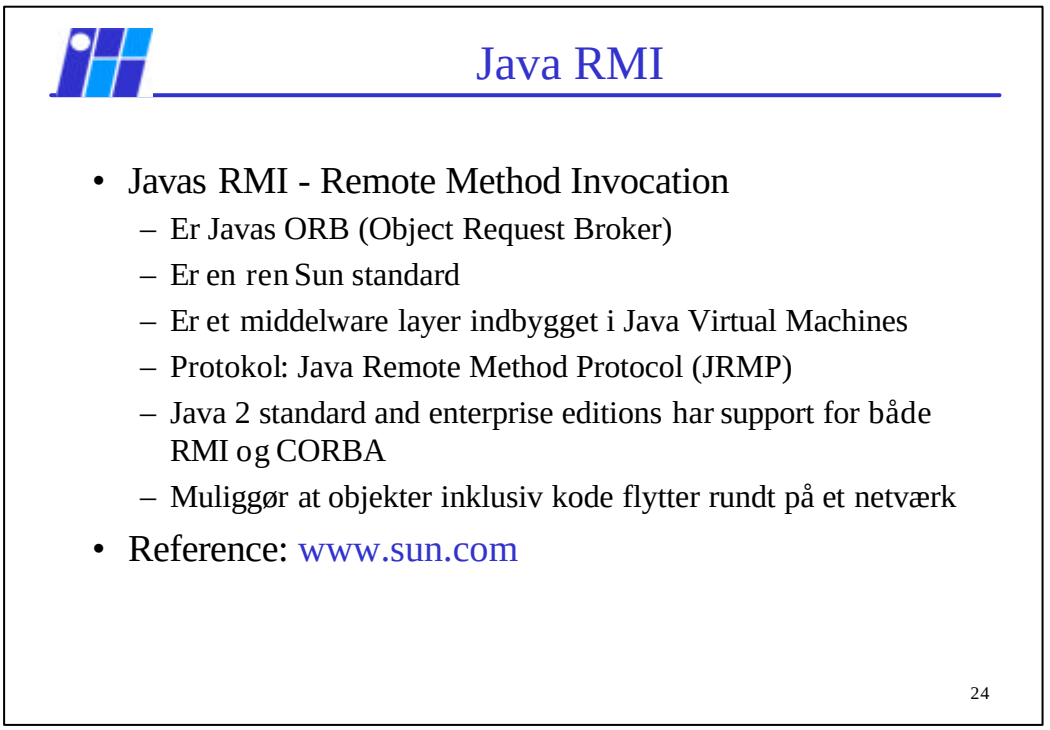
21

CORBA til apparater

- Minimum CORBA for embedded system 1998
  - en neddroslet version f.eks. uden *Dynamic Invocation if.*
- Real-Time CORBA standard 1999
  - udvidelser: prioritetsbegreb, threadpool, konfigurering og valg mellem forskellige protokoller
- ORB til embedded systemer
  - Et eksempel: OrBacus/E på Windows 2000/PC (www.iona.com)
    - C++ "Hello world" applikation: client 98 Kb, Server 164 Kb
    - Java "Hello world" applikation; client 63 Kb, Server 92 Kb
    - Embedded Event services: 270 Kb, Naming Services: 260 Kb
- Anvend et bibliotek *IIOP engine*
  - tillader at man anvender IIOP protokollen (fås ned til 15 Kb)

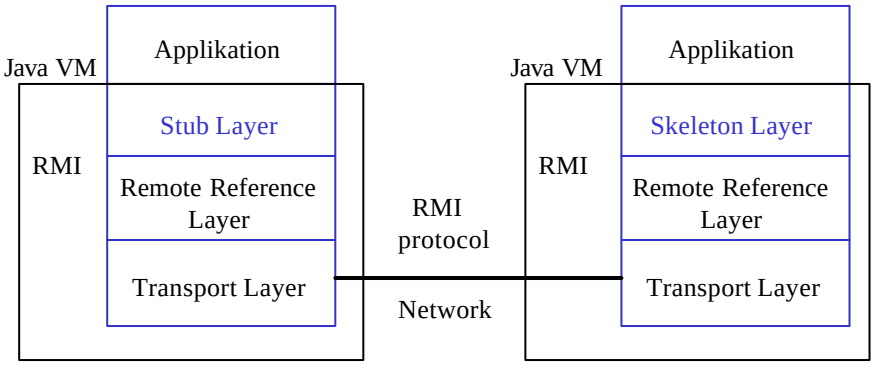
22


Eksempel på anvendelse af CORBA
Hovedcomputer
Web browser
Java
User Interface
ORB
upload
Device 1
http
HTML+
Java
applet(s)
ORB
C++
applikation
Eksempler på applets:
• til konfigurering
• til aflæsning af data
upload
Alternativ:
En central Web server med
applets for devices
Device 2
ORB
C++ / C
applikation
23


Java RMI
• Javas RMI - Remote Method Invocation
– Er Javas ORB (Object Request Broker)
– Er en ren Sun standard
– Er et middelware layer indbygget i Java Virtual Machines
– Protokol: Java Remote Method Protocol (JRMP)
– Java 2 standard and enterprise editions har support for både RMI og CORBA
– Muliggør at objekter inklusiv kode flytter rundt på et netværk
• Reference: www.sun.com
24

Java RMI arkitektur

RMI er indeholdt i Javas Virtual Machine (VM)



25

Java RMI styrker/svagheder

- Styrker:
  - Er ren Java - en styrke i et Java miljø
  - Kan samarbejde med CORBA via RMI-IIOP
  - Følger med som en integreret del af Java
  - Kan overføre objekter inklusiv koden for operationerne
  - Håndterer *remote garbage collection*
- Svagheder:
  - Er ren Java teknologi, men kan dog samarbejde med CORBA
  - Virker ikke godt gennem firewalls
  - Kræver nogen erfaring at anvende (dog lettere end CORBA)

Kilde: Mike Preradovic, Intrinsic Software, ESC2001

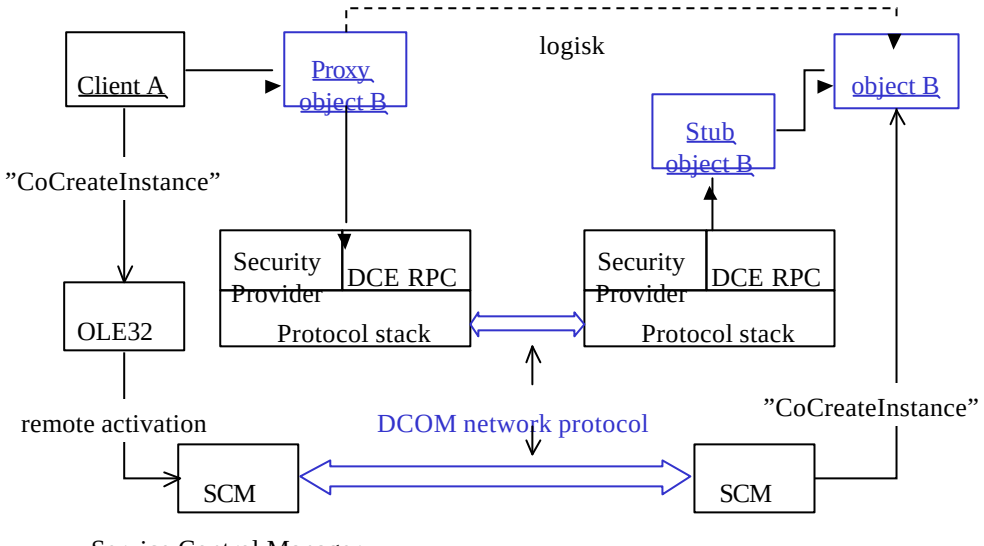
26

DCOM

- Microsoft standarderne COM og DCOM:
  - COM - Component Object Model
  - DCOM - Distributed COM
    - released i 1997, Microsofts version af CORBA
  - DCOM er integreret i alle Window platforme
    - incl. Windows CE 3.x og supporteres på andre platforme f.eks. Unix, Linux
  - Protocol: Object Remote Procedure call (ORPC)
  - Er en binær standard
    - Interfaces specificeres med et memory layout svarende til C++'s vtable (virtual method table)
  - Kan anvendes fra f.eks. C++, Delphi, Visual Basic og Java
- Reference: www.microsoft.com/com/dcom.htm

27

DCOM arkitektur



28

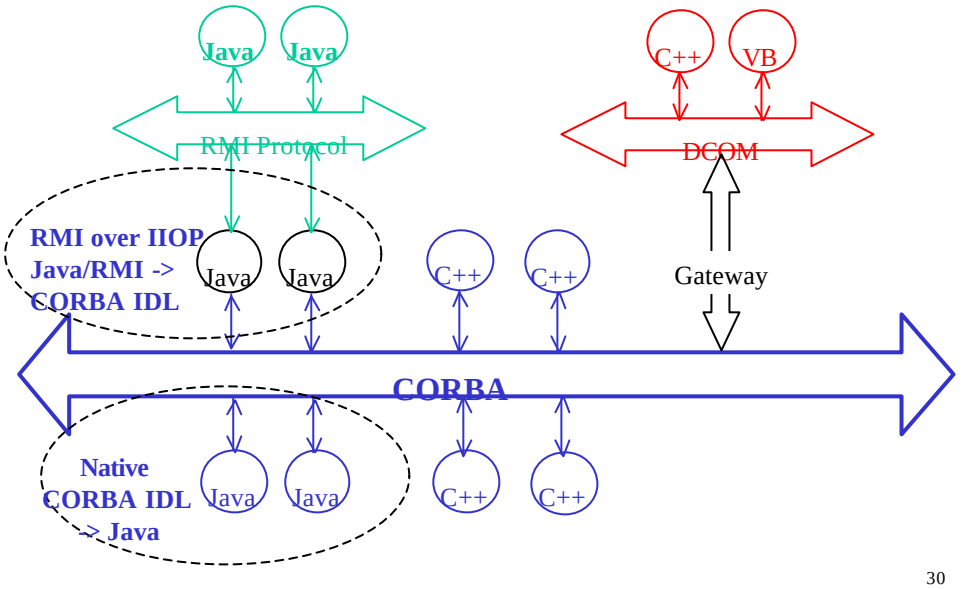
DCOM styrker/svagheder

- Styrker:
  - Er på alle Microsoft platforme og er anvendt i industrien i flere år
  - Er sproguafhængig
  - Mange Microsoft applikationer er DCOM enabled
  - DCOM er gratis
- Svagheder:
  - Er en Microsoft standard
  - Er ikke Java venlig
  - Er meget kompleks at programmere i C++
  - Har ingen reelle services som f.eks. CORBA
  - Egner sig ikke til indlejrede decives pga. størrelsen
  - Virker ikke godt gennem firewalls
  - Dårlig opførsel i ”disconnected” environments

Kilde: Mike Preradovic, Intrinsic Software, ESC2001

29

Interworking: CORBA, RMI og DCOM



30

Auto-discovery teknologier

- Enheder der skal kobles på et netværk bør fremover overveje at have funktionalitet, der muliggør *auto discovery* og *auto configuration*
- Der er pt. flere konkurrende *defacto* standarder:
  - UPnP (Universal Plug and Play)
  - Jini
  - Salutation
  - HAVI
- Disse er pt. ikke særligt udbredte

31

UPnP

- UPnP: Universal Plug and Play - oprindelig en Microsoft standard
  - Version 1.0 af arkitekturen er defineret
  - Support for UPnP i Windows Me - ikke pt. i Window CE
  - UPnP forum: konsortium med 200 leverandører
- Defineret som standardudvidelser til TCP/IP, DHCP og andre Internet standarder
- *Discovery* via en decentral *broadcast* process
- Data udveksling foregår via XML
- Kontrol via SOAP
- Reference: www.upnp.org

32

XML remote call/object modelling

- XML (eXtensible Markup Language)
  - Er en World Wide Web Consortium (W3C) standard
  - Anvendes til at repræsentere komplekse data i et læsbart tekstformat
- Reference: www.w3.org/XML/
- Pt. er der 3 XML baserede distribuerede skemainitiativer:
  - SOAP (Simple Object Access Protocol) - Microsoft
  - XML/RPC - en åben standard for modellering af RPC kald som XML
  - Et OMG initiativ for anvendelse af XML sammen med CORBA

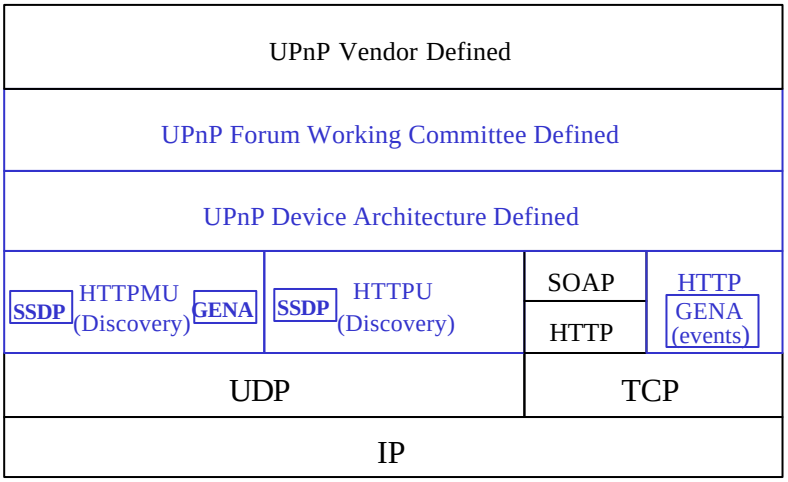
33

SOAP

- SOAP: (Simple Object Access Protocol)
  - Definerer anvendelsen af XML og HTTP til at eksekvere remote procedure call (RPC)
  - En standard for RPC over internettet
  - Kan arbejde effektivt med proxier og firewalls
  - Kan anvende Secure Socket Layer (SLL) for sikkerhed
  - Hvert UPnP control request samt retursvar er en SOAP message
- SOAP og XML kan anvendes uafhængig af UPnP

34

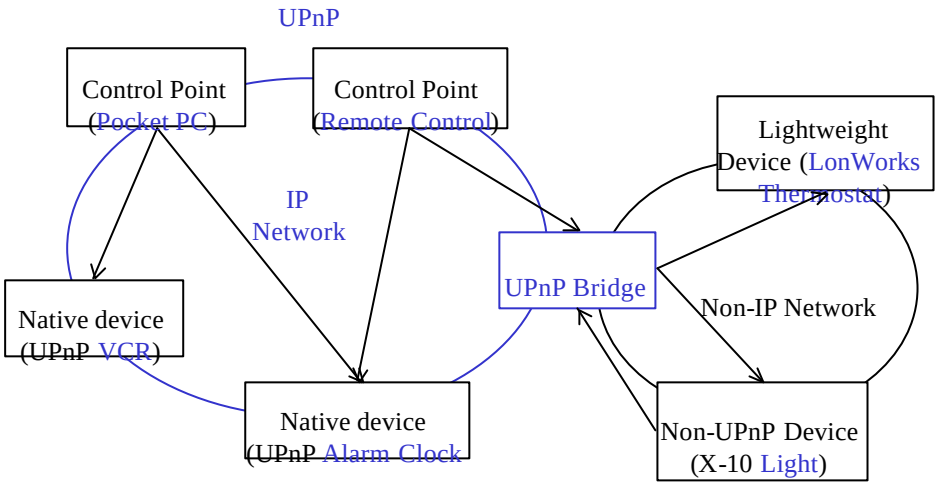
UPnP arkitektur



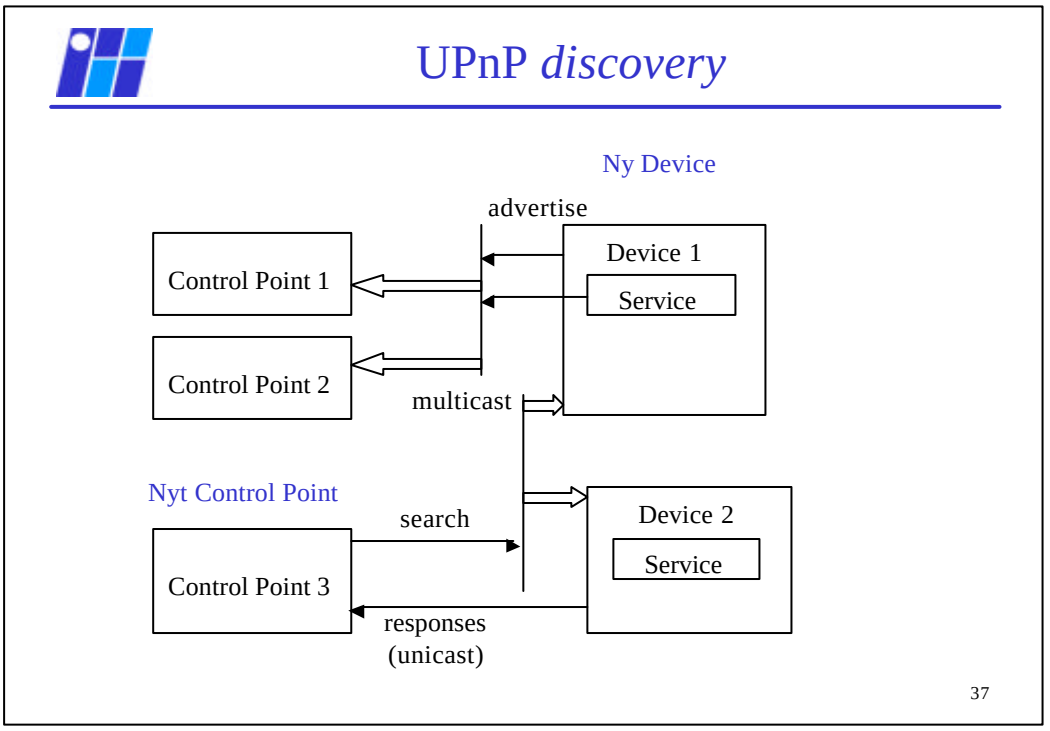
SSDP: Simple Service Discovery Protocol
GENA: Generic Event Notification Architecture

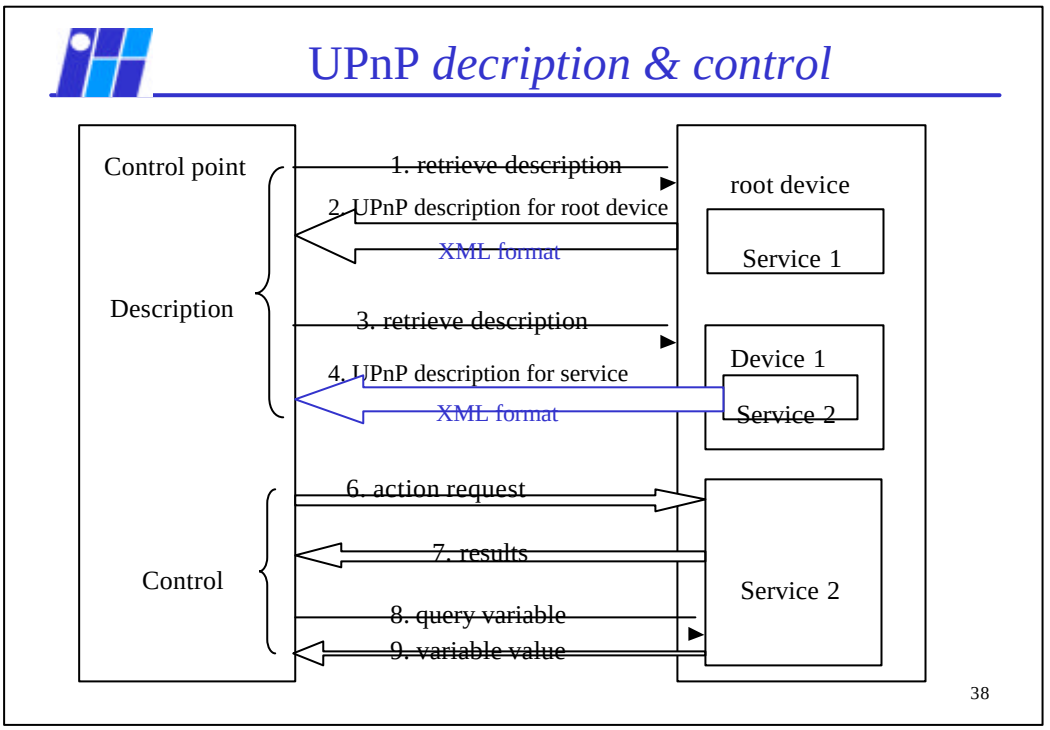
35

UPnP - eksempel

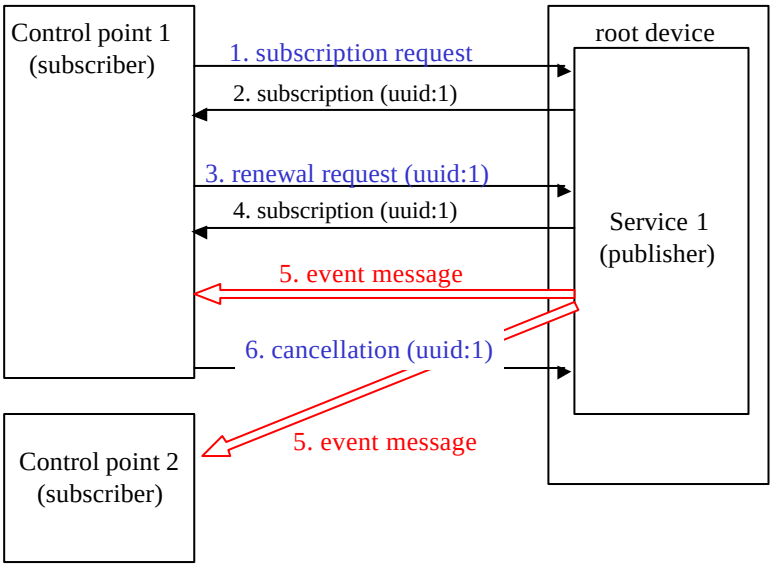


36


UPnP discovery
Ny Device
advertise
Control Point 1
Device 1
Service
Control Point 2
multicast
Nyt Control Point
search
Device 2
Service
Control Point 3
responses
(unicast)
37

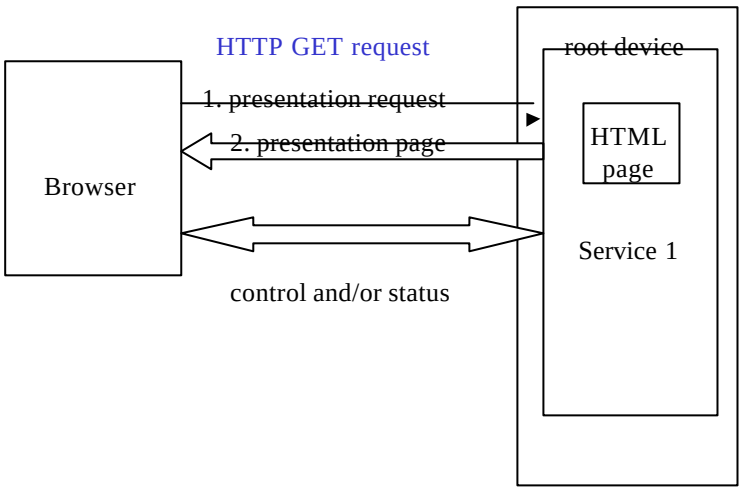

UPnP decription & control
Control point
1. retrieve description
root device
2. UPnP description for root device
XML format
Service 1
Description
3. retrieve description
Device 1
4. UPnP description for service
XML format
Service 2
6. action request
7. results
Control
Service 2
8. query variable
9. variable value
38

UPnP *eventing*



39

UPnP *presentation*



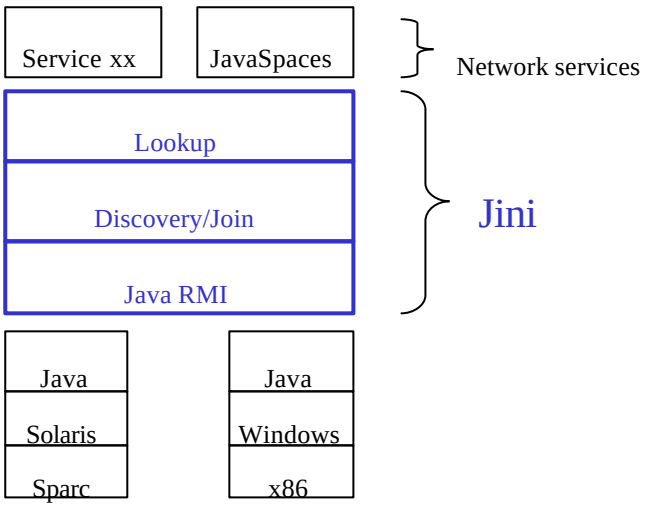
40

Jini

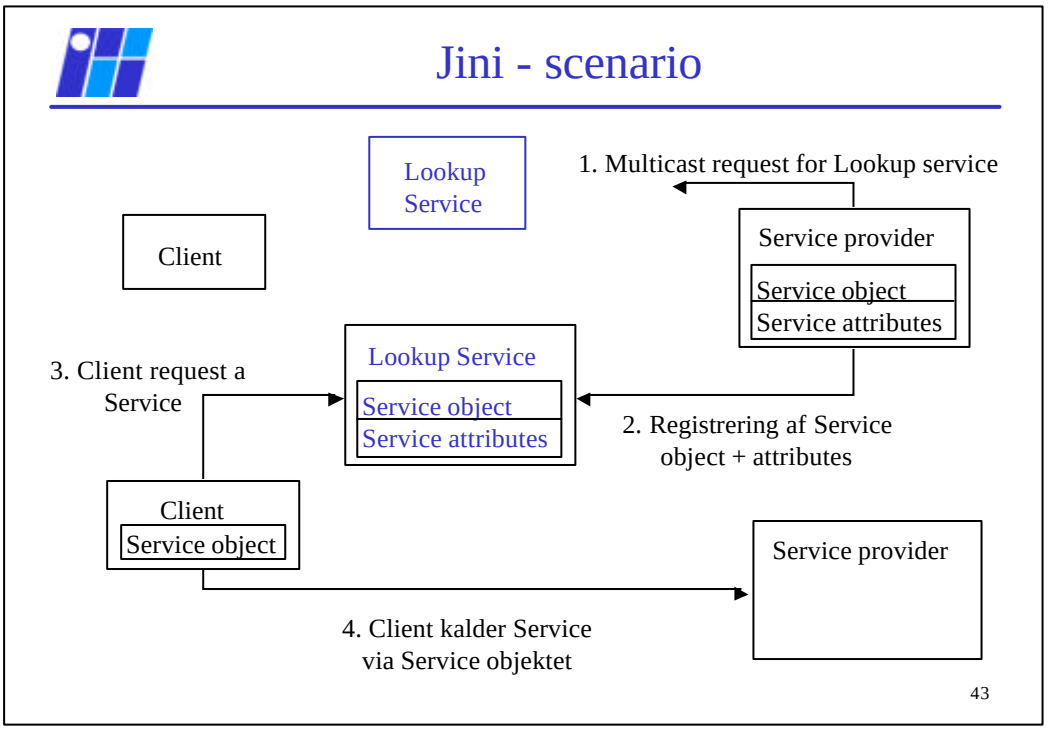
- Jini er en SUN standard
- Jini er en ren Java teknologi, der kræver:
  - Java 2 (JDK 1.2) eller højere
  - Java's Remote Method Invocation (RMI)
- Jini kræver at netværket og enhederne supporterer IP mulitcasting
- Jini supporterer distribuerede events med notifications
- En "device" kan f.eks registrere en brugerdialog til styring af "devicen" som en service
- Reference: www.jini.org

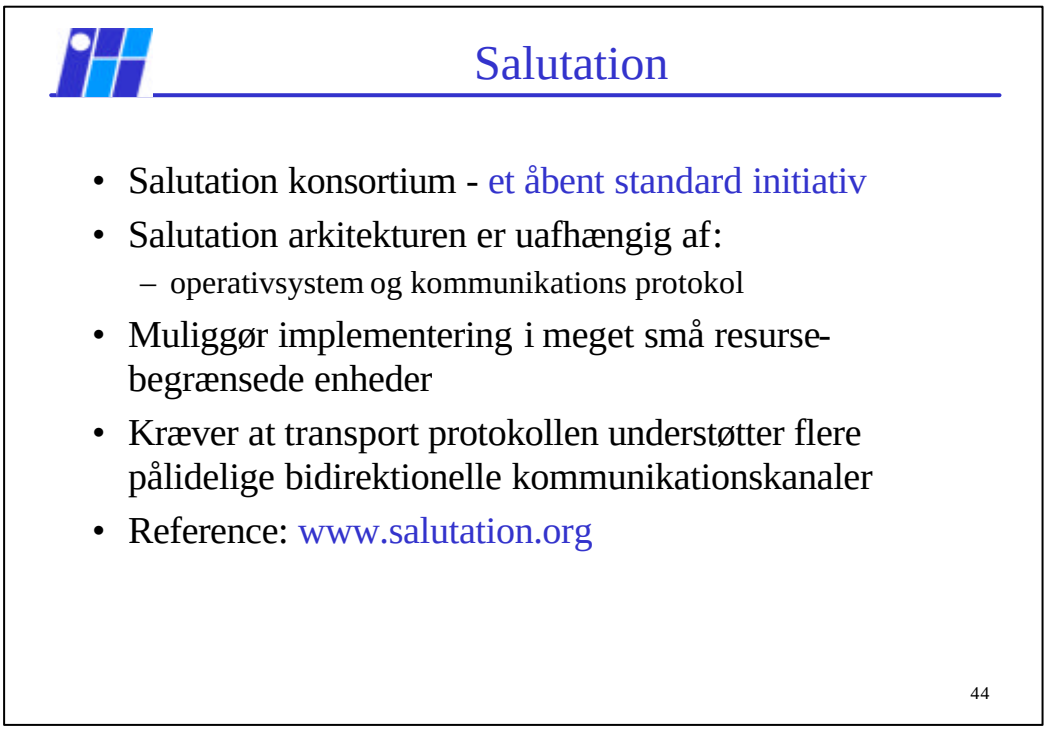
41

Jini arkitektur

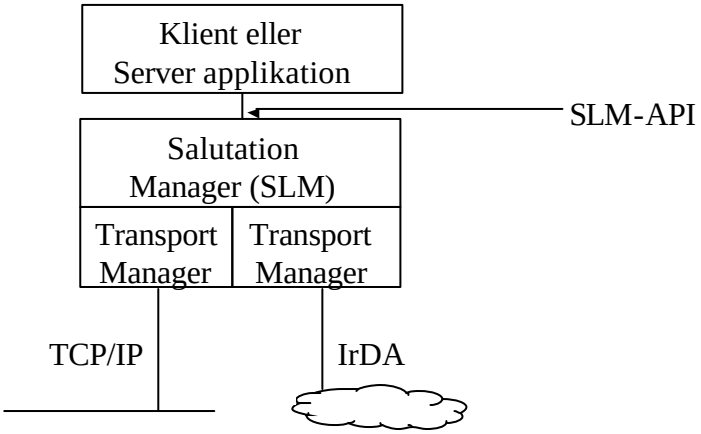


42


Jini - scenario
Lookup Service
1. Multicast request for Lookup service
Client
Service provider
Service object
Service attributes
Lookup Service
Service object
Service attributes
3. Client request a Service
2. Registrering af Service object + attributes
Client
Service object
Service provider
4. Client kalder Service via Service objektet
43


Salutation
• Salutation konsortium - et åbent standard initiativ
• Salutation arkitekturen er uafhængig af:
– operativsystem og kommunikations protokol
• Muliggør implementering i meget små resurse-begrænsede enheder
• Kræver at transport protokollen understøtter flere pålidelige bidirektionelle kommunikationskanaler
• Reference: www.salutation.org
44

Salutation arkitektur

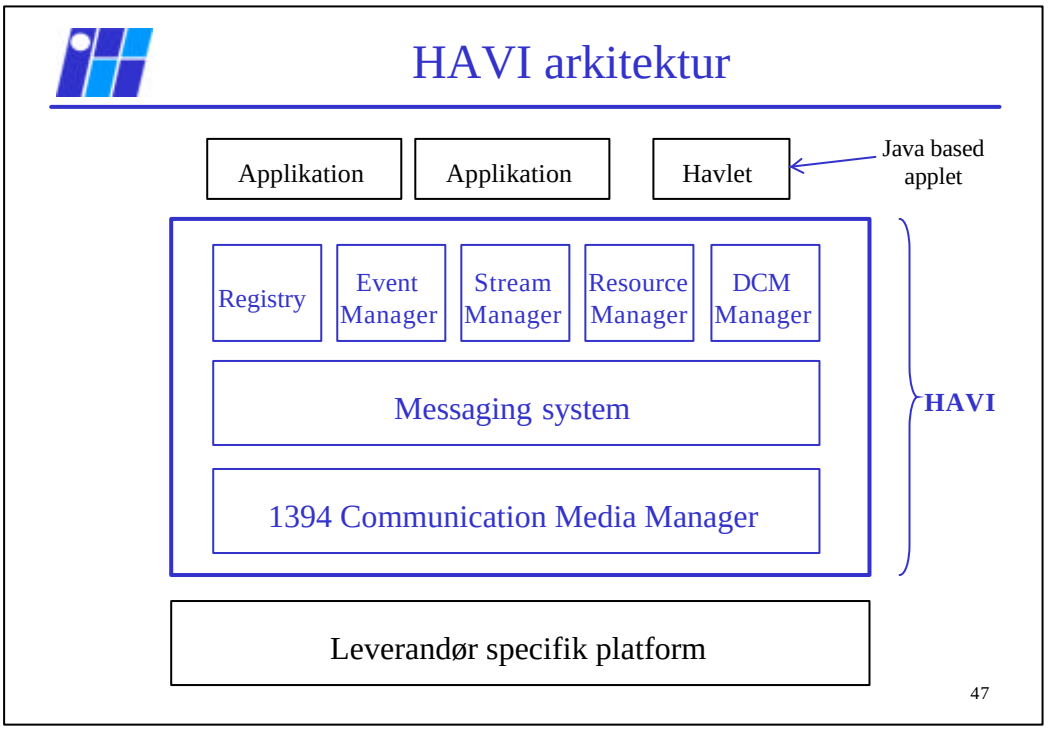


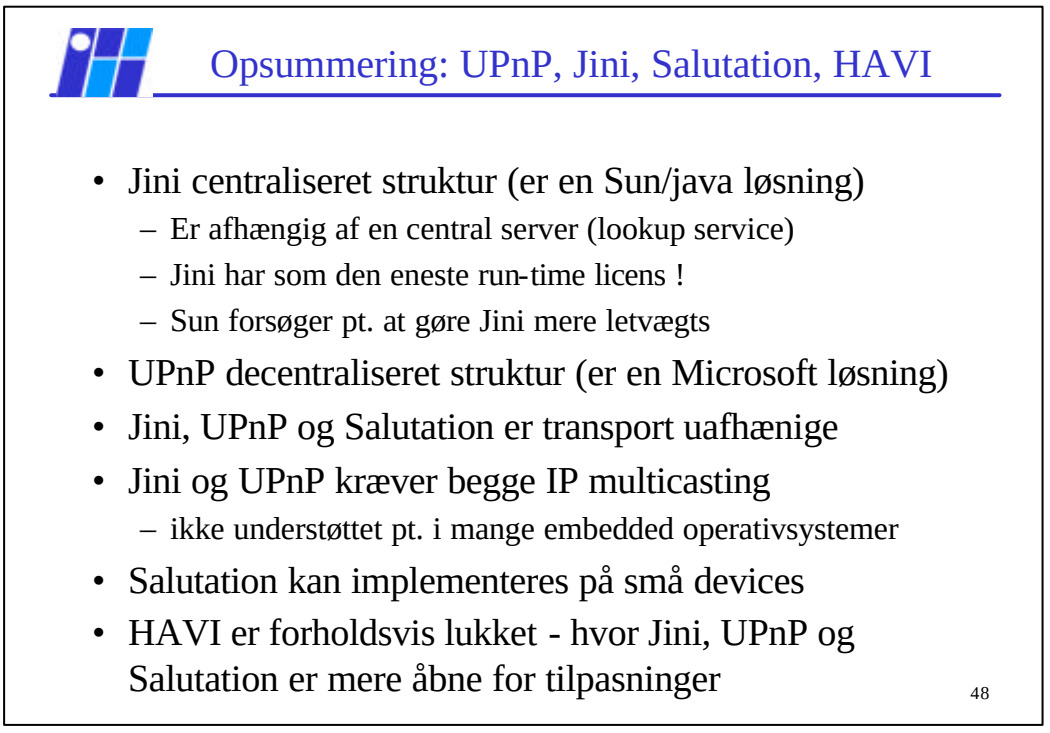
45

HAVI

- HAVI - Hardware Audio/Video Interoperability
  - Version 1.0 af specifikationen er released
  - Defineret af Grundig, Hitachi, Matsushita, Philips, Sharp, Sony, Thomson og Toshiba
- Defineret ovenpå standarden IEEE-1394 (Firewire)
- Anvender 1394's *device & service discovery* mekanisme
- Definerer et statisk kontrol interface for enheder
- Der eksisterer *bridges* til Jini
- Bridges til UPnP er under overvejelse
- Reference: www.havi.org

46


HAVI arkitektur
Applikation
Applikation
Havlet
Java based applet
Registry
Event Manager
Stream Manager
Resource Manager
DCM Manager
Messaging system
1394 Communication Media Manager
HAVI
Leverandør specifik platform
47


Opsummering: UPnP, Jini, Salutation, HAVI
• Jini centraliseret struktur (er en Sun/java løsning)
– Er afhængig af en central server (lookup service)
– Jini har som den eneste run-time licens !
– Sun forsøger pt. at gøre Jini mere letvægts
• UPnP decentraliseret struktur (er en Microsoft løsning)
• Jini, UPnP og Salutation er transport uafhænige
• Jini og UPnP kræver begge IP multicasting
– ikke understøttet pt. i mange embedded operativsystemer
• Salutation kan implementeres på små devices
• HAVI er forholdsvis lukket - hvor Jini, UPnP og Salutation er mere åbne for tilpasninger
48

Opsummering

- Der er en rivende udvikling inden for distribueret kommunikation og middelware
- Der er gode, brugbare men desværre konkurrende standarder
- Standarderne er pt. "store" i forhold til apparatsystemer men på vej i mindre og mere RT orienterede udgaver
- Automatisk tilslutning og konfigurering bliver et must for nogle typer af apparater
  - Der er her tre generelle bud (Jini, UPnP, Salutation) samt HAVI
- XML forventes at blive anvendt til udveksling af kontrol- og statusinformation til apparater

49

Samarbejdsmuligheder med IHA

- Afgangsprojekter på diplomingeniørstudiet
  - udført i virksomheden
  - eller på IHA
  - vejleder fra virksomheden eller fra IHA
  - omfang 2/3 af sidste semester (20 ECTS point)
- Ingeniørpraktik – ½ år på 5 semester
- På længere sigt – specialeprojekter på overbygningsuddannelsen (start forår 2003)
- Eksterne lektorer
- Eksterne censorer (udpeges for 4 årig periode)
- Vi søger også nye kollegaer til IKT linien og til overbygningsuddannelse

50

Referencer

- CORBA: www.omg.org
- RMI: www.sun.com
- DCOM: www.microsoft.com/com/dcom.htm
- Jini: www.jini.org
- UPnP: www.upnp.org
- Salutation: www.salutation.org
- HAVI: www.havi.org
- XML: www.w3.org/XML/
- **www.esconline.com/papers.htm**
  (registrer, select conf: SF2001, papers 411,420, 325)

NY ift. handout

51