

JOHDANTO

Johdanto - agenda

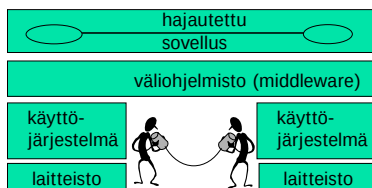
- Mitä väliohjelmistolla tarkoitetaan?
- Väliohjelmistojen rooli ja tehtävät
- Väliohjelmiston haasteet ja syntymekanismit
- Väliohjelmistojen kehitysvaiheet – haasteet ja trendit
- Väliohjelmistojen kategorisointia

11.9.2007

Väliohjelmistot - Lea Kutvonen

2

Mitä väliohjelmistolla tarkoitetaan?



11.9.2007

Väliohjelmistot - Lea Kutvonen

3

Middleware definition 1

Software that is used to move information from one program to other program in a distributed environment, shielding the developer from dependencies on communication protocols, operating systems, and hardware platforms.

Campbell et.al.

11.9.2007

Väliohjelmistot - Lea Kutvonen

4

Middleware definition 2

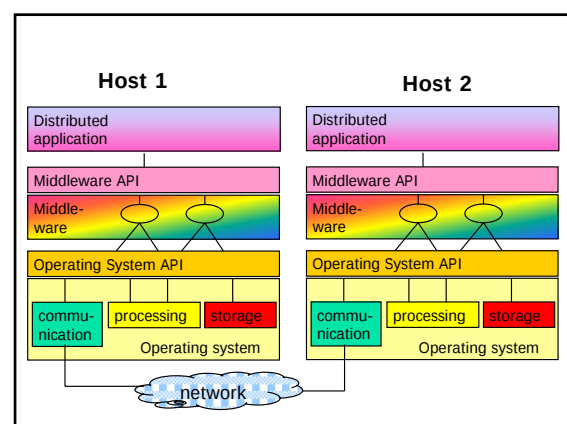
Middleware is a class of software technologies designed to help manage the complexity and heterogeneity inherent in distributed systems. It is defined as a layer of software above the operating system but below the application program that provides a common programming abstraction across a distributed system.

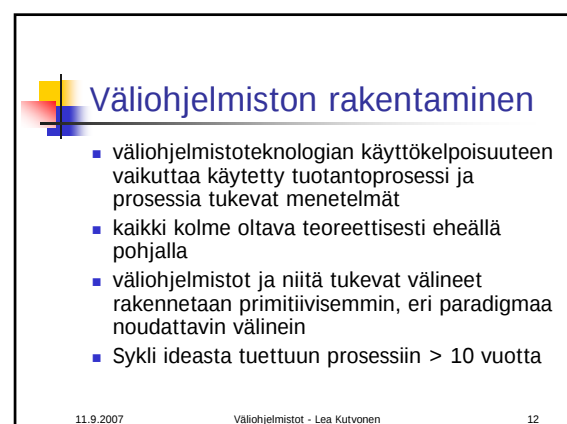
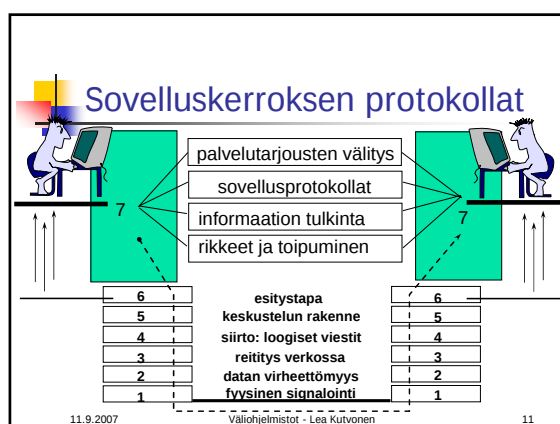
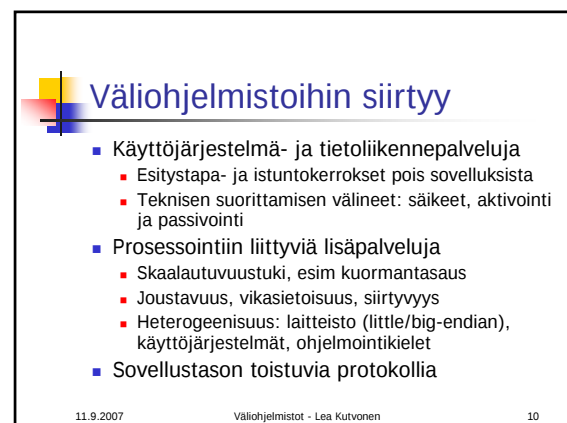
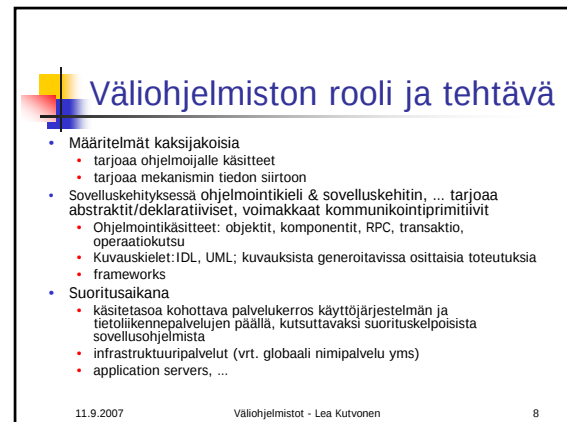
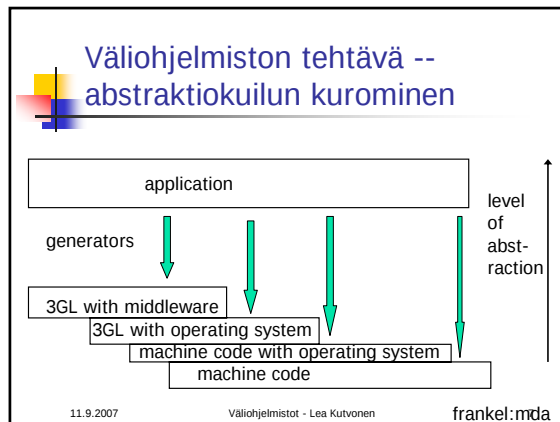
Bakken 2001

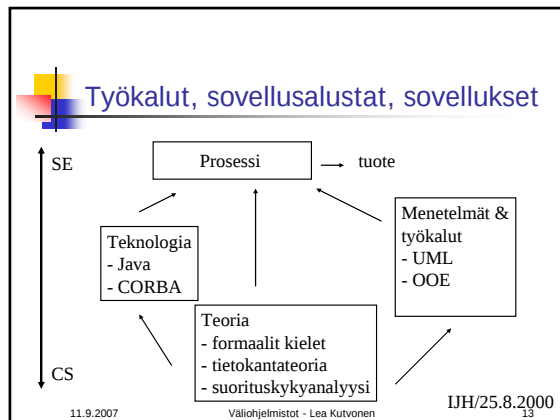
11.9.2007

Väliohjelmistot - Lea Kutvonen

5







Työkalut, sovellusalustat, sovellukset

- Toteutus- ja suoritusaikaiset maailmat eivät erillisiä vaan tukevat toisiaan
- esimerkiksi nykyisissä adaptiivisissa sovelluksissa
 - Tavoite: Adaptiiviset sovellukset muuntuvat suoritusaikana
 - Edellytys: Komponenttien kyky tulla korvatuksi toisilla
 - Ohjelmointikäsité: Myöhäinen sidonta (late binding)
 - Infrastruktuuripalvelu: Palvelujen paikantaminen ominaisuuksien perusteella (meklaus)

11.9.2007 Väliojelmistot - Lea Kutvonen 14

Arkkitehtuurimallit virittämässä väliojelmiston tukemia käsitteitä

- Architectural style - architectural *design pattern*
- system = *components & connectors*
 - Component: client, server, filters, layers, databases
 - Connector: procedure call, event broadcast, database, protocols, pipes
- some styles
 - Dataflow (Pipes and filters, Batch sequential)
 - Data-centered (Repository)
 - Virtual Machine (Interpreter, Rule-based system)
 - Call and Return (Main program and subroutine, Object-oriented, Layered)
 - Independent Components (Communicating processes, Client/server, Event systems with implicit or explicit invocation)

11.9.2007 Väliojelmistot - Lea Kutvonen 15

Arkkitehtuurimallit virittämässä väliojelmiston tukemia käsitteitä

- Example: Event-based, implicit invocation
- Component: (active) object, capsule, module, ...
 - Can be an instance of a class, an active class, or simply a module (non-OO).
 - Have at their interface, methods and ports.
- Connector: "connector", channel, binding, callback.
 - Offers one-to-one, one-to-many, many-to-one connections;
 - Asynchronous event broadcast.
 - (Synchronous event post & await reply => call-and-return.)

11.9.2007 Väliojelmistot - Lea Kutvonen 16

Väliojelmiston valinnasta

- Mikä on sovellusalueen kannalta luonteva arkkitehtuurityyppi?
- Millainen kommunikointisemantiikka tarvitaan? Synkronisuus, transaktionaalisuus, tietovirta, monilähetys, globaali tietovarasto, portaali-taustajärjestelmä, ...
- Mitä järjestelmäpalveluita kommunikointisemantiikan tukemiseen on tarjolla? Entä komponentin pysyvyyden, rinnakkaisuuden, tilallisuuden yms tukemiseen?
- Mitä ohjelmointikielitasen ilmaisuja voidaan käyttää?
- Mitä lisäpalveluita kääntäjistä, kirjastoista, kehittimistä, olio- tai komponenttikehyksistä on saatavissa?

11.9.2007 Väliojelmistot - Lea Kutvonen 17

Väliojelmistojen kehitysvaiheet

The diagram shows the evolution of middleware. It starts with 'EAI' (Enterprise Application Integration) and 'multimedia'. It then moves to 'adaptiivisuus' (Adaptability) and 'web services'. It then moves to 'QoS' (Quality of Service), 'PBM/WF' (Business Process Management/Workflow), and 'mobiilit' (Mobility). It then moves to 'komponentit' (Components) and 'B2Bi' (Business-to-Business Integration). It then moves to 'objektiväylät' (Object-oriented middleware). It then moves to 'yleiskäyttö-väliojelmisto' (General-purpose middleware). It then moves to 'tapahtuma-monitorit' (Event monitors), 'hajautettu kj standardirajapinta, standardiprotokollat' (Distributed object standard interface, standard protocols), and 'RPC' (Remote Procedure Call). It then moves to 'tietokannan etäkäyttö' (Database remote access).

11.9.2007 Väliojelmistot - Lea Kutvonen 18

Väliohjelmisto-termin synty

- termi syntyi 1980-luvun lopulla
- vielä 1990-luvun puolivälissä
 - väliohjelmisto-termi liitettiin etupäässä relaatiotietokantoihin
 - nykyiset väliohjelmistot kulkivat nimillä network operating systems, distributed operating systems, distributed computing environments
- merkittäviä järjestelmiä
 - Cronus, Clouds, Eden
 - RPC by Birel & Nelson -84, ONC (SUN), NCS (Apollo), DCE (X/Open)

11.9.2007

Väliohjelmistot - Lea Kutvonen

19

Väliohjelmistojen alkuajat

- Hajautetun käyttöjärjestelmän tehtävänä tuottaa illuusio yhdestä tietokoneesta
 - Hajautettu prosessien hallinta
 - Prosessien elinkaari, allokointi,
 - prosessien välinen kommunikointi ja synkronointi
 - RPC, viestit
 - Hajautettu resurssien hallinta
 - resurssien varaus, suojaus
 - lukkiutumien havaitseminen
 - Hajautetut palvelut
 - hajautettu tiedostojärjestelmä, muisti
 - hierarkkinen globaali nimentä

11.9.2007

Väliohjelmistot - Lea Kutvonen

20

Haasteita ja ratkaisutapoja

- Haasteita järjestelmäarkkitehtuurille
 - Yrityksen tietojärjestelmän integrointi
- Haasteita järjestelmien toteuttajille
 - Ohjelmiston siirto uuteen järjestelmään
- Ratkaisumalli:
 - Standardoidut ohjelmointirajapinnat
 - Ohjelmointikielen tuettava peruspalveluita (kutsut jne)
 - Kielen ulkopuolella järjestelmärajapinnat (tietokannat, protokollien käyttö, jne)
 - Standardoidut protokollat

11.9.2007

Väliohjelmistot - Lea Kutvonen

21

Väliohjelmistot vakiintuvat

- 1990-luvun loppupuolella väliohjelmisto erottuu muista ratkaisuista
 - yleiskäyttöinen (vrt. sovellusaluekohtainen framework)
 - tarjoaa hajautettuja palveluja
 - eri laitteisto-, kj- ja verkkoympäristöihin toteutettu
 - tukee standardoituja siirtoprotokollia ja tarjoaa standardi-API:n
- merkittäviä järjestelmiä
 - DCE, COM, CORBA

11.9.2007

Väliohjelmistot - Lea Kutvonen

22

Haasteita ja ratkaisutapoja

- Haasteita järjestelmäarkkitehtuurille ja järjestelmien toteuttajille
 - Erillisten sovellusten yhteistoiminta yrityksen sisällä
 - asiakasportaalit
- Ratkaisumalli:
 - Standardoidut muodot tiedon esitystavalle
 - Standardoidut väliohjelmistorajapinnat
 - Standardoidut tavat esitellä sovelluksia käytettäväksi
 - Sovelluspalvelimet (application servers), "hotellit"

11.9.2007

Väliohjelmistot - Lea Kutvonen

23

Perinteiset väliohjelmistot

- Perusmalleja
 - Tietokantojen hajautus, tapahtumamonitorit
 - Hajautettu kj tai verkkokj
 - Objektiväylät, objektikielten toteutus
- Sovellusalueita
 - Pankkijärjestelmät
 - Sairaalan tietojärjestelmä
 - Verkkopalvelut ja dokumenttien siirto

11.9.2007

Väliohjelmistot - Lea Kutvonen

24

Väliohjelmistot nykyisin

- Vaativimmat sovellusalueet
 - Multimediajärjestelmät
 - Tietovirtojen siirtäminen (streams) jatkuvaa valvontaa vaativaa
 - Reaaliaikajärjestelmät
 - Reaaliaikaskedulointi, sopimukset, resurssienvaraus
 - Telecom
 - Vikasietoisuus, järjestelmien suoritusaikainen päivittäminen
 - Sähköinen liiketoiminta
 - Järjestelmien globaalisuus
 - Turvallisuusnäkökohdat avoimessa verkossa
 - Adaptiiviset järjestelmät
 - Personoitavissa, paikkatietoiset, laitteistoon sopeutuvat, ubiquitous computing

11.9.2007 Väliohjelmistot - Lea Kutvonen 25

Väliohjelmistot nykyisin

- Nykyaikainen väliohjelmisto tarjoaa standardoidut välineet käsitellä prosessointiyksiköitä ja niiden välistä vuorovaikutusta
- Aikaisempaa abstraktimpia ja voimakkaampia ohjelmointikäsitteitä
 - objektit ja komponentit
 - palvelun kuvaus ja toteuttaja erotetaan
 - metadatan käyttö
 - sijainti- ja pääsytuntumattomuus, muut tuntumattomuuspalvelut

11.9.2007 Väliohjelmistot - Lea Kutvonen 26

Väliohjelmistot nykyisin

- Objektit ja komponentit
 - Loogiset vs. fyysiset
 - Miten kuvautuvat lopulta suorittaville prosesseille? Granulariteetti?
 - Kommunikointimuodot, rajapintojen lkm?
 - Mallin odottama kommunikointituki?
 - Myöhäinen sidonta
 - Miten loogista objektia vastaavaa fyysistä konfiguraatiota hallitaan?

11.9.2007 Väliohjelmistot - Lea Kutvonen 27

Väliohjelmistot nykyisin

- palvelun kuvaus ja toteuttaja erotetaan
 - Tarpeita sekä suoritus- että toteutusaikana
- Kommunikoinnin laatu -> eriytetyt tuntumattomuuspalvelut

11.9.2007 Väliohjelmistot - Lea Kutvonen 28

Väliohjelmistot nykyisin

- metadatan käyttö
 - Nimet yms. paikantamistavat
 - Palvelun laatu –sopimukset (QoS jne)
 - Informaation rakenne
 - Rajapinta-, komponentti- ja palvelukuvaukset
 - Ohjelmistoarkkitehtuuri, konfiguraatiot

11.9.2007 Väliohjelmistot - Lea Kutvonen 29

Haasteita ja ratkaisutapoja

- Haasteita järjestelmien toteuttajille
 - Laajan ohjelmiston tuottaminen tai koostaminen
- Ratkaisumalli:
 - Informaation käsittely: data+metadata
 - Standardoitu, adaptiivinen väliohjelmistorajapinta
 - Ilmaisuvoimaiset välineet kuvata palveluja
 - Toteutusten osittainen generointi palvelukuvauksista

11.9.2007 Väliohjelmistot - Lea Kutvonen 30

Väliohjelmistojen tulevaisuus

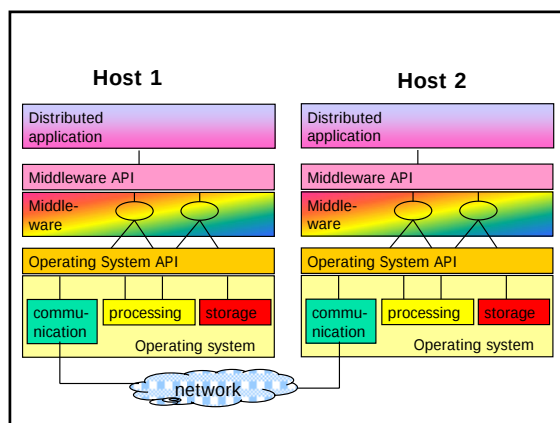
- Haasteita järjestelmäarkkitehtuureille
 - Organisaatorajat ylittävä palvelujen integrointi
- Järjestelmien toteuttajille
 - Laajan ohjelmiston koostaminen
 - Palvelumarkkinoiden toteuttaminen
- Ratkaisumalli:
 - Standardoitujen metainformaatiopalveluiden käyttö
 - Globaali arkkitehtuurimalli?
 - Erilaiset arkkitehtuurityylit ja komponenttimallit yhteensovitettavissa?

11.9.2007 Väliohjelmistot - Lea Kutvonen 31

Väliohjelmistotyyppejä

- Tapahtumaorientoitu vo. (Transactional mw)
 - IBM CICS, BEA Tuxedo, Transarc Encina
- Viestiperustainen vo. (Message oriented middleware, MOM)
 - IBM MQSeries, Sun Java Message Queue
- Proseduuripohjainen vo. (Procedural mw)
 - RPC kaikkine variaatioineen
- Objekti- ja komponenttipohjainen vo.
 - CORBA, DCOM, Java RMI, EJB, SOAP, .NET

11.9.2007 Väliohjelmistot - Lea Kutvonen 32



Väliohjelmistotyyppejä

- eri tyypit ottavat eri aspektit huomioon
 - distributed tuples: communication, storage, limited in processing
 - MOM: communication, no processing, limited storage
 - RPC: communication, processing, no storage
 - distributed objects: communication, processing, and storage

11.9.2007 Väliohjelmistot - Lea Kutvonen 34

