

Tietojärjestelmäarkkitehtuurit

ITK130 Johdatus ohjelmistotekniikkaan
Syksy 2003

Sami Kollanus

Aluksi...

- Tietojärjestelmäarkkitehtuurit vs. ohjelmistoarkkitehtuurit
- Pohjana Tietojärjestelmäarkkitehtuurit –kurssi:
 - <http://www.cs.jyu.fi/~heikki/TJTL77.html>
- Tulevaisuudessa seuraavat syventävät kurssit:
 - Yrityksen tietojärjestelmien integrointi 4 ov (tietohallinto)
 - Ohjelmistoarkkitehtuurit 3 ov (ohjelmistotekniikka)

Lähtökohta yrityksessä

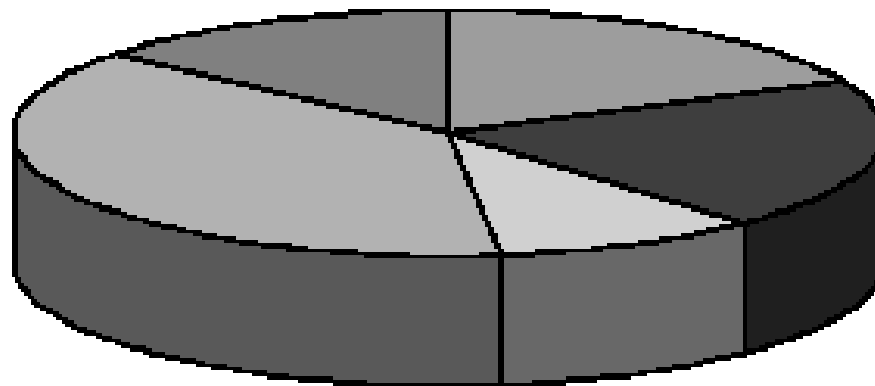


Miksi integroida?

- Because we can?
- Softan tehtävä on TUKEA liiketoimintaa!
- Toiminnan tehostaminen
- Kommunikoinnin tehostaminen

- Säästö resursseissa
- Kilpailuetu palvelussa
 - Tarjotaan tietoa, jota muut ei tarjoa
 - Nopeus

The Need for Application Integration



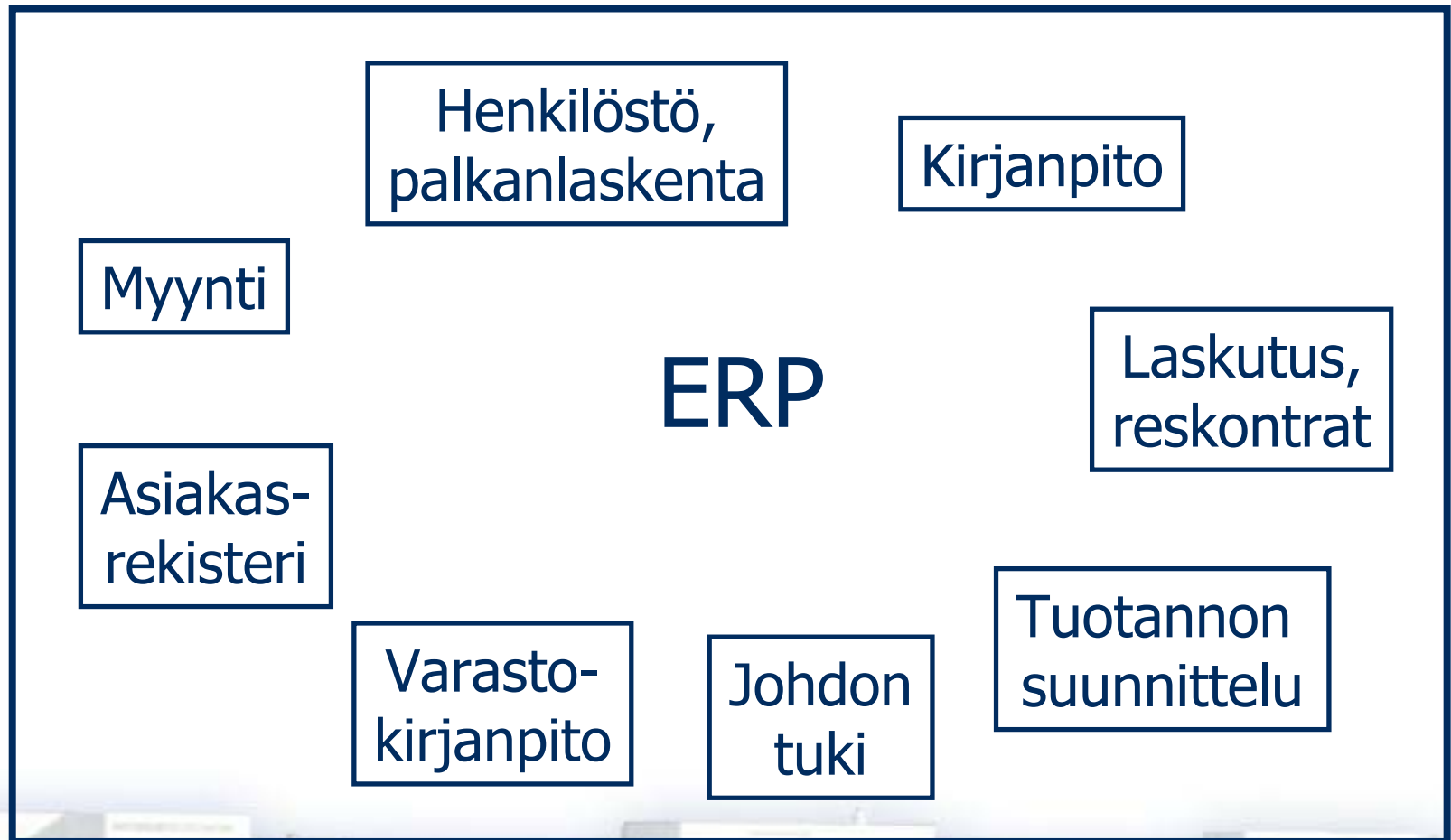
- Expand Supplier and Customer Relationships
- Move information beyond traditional confines
- Mergers and acquisitions
- Get new applications into production quickly
- Integrate a Best-of-Breed world

Source: Aberdeen Group, October 1998

Ongelmia

- Vanhat järjestelmät:
 - Liittymien rakentamiseen suuri osa työstä
 - Monia kriittisiä järjestelmiä, joita ei taloudellisesti kannata päivittää
 - Yhteensovittaminen uusiin järjestelmiin
- Pakettiohjelmien liittymät ovat huonot
- Fyysisesti eri paikoissa olevat sovellukset ja tietokannat

Toiminnanohjausjärjestelmä (ERP)



ERP business

Suurimmat softafirmat

1. Microsoft
- 2. Oracle**
- 3. SAP**
4. Computer Associates
5. Softbank
6. Compuware
7. Siebel Systems
- 8. PeopleSoft**
9. BMC Software
10. Electronic Arts

Suurimmat ERP-toimittajat (2001)

1. SAP
2. PeopleSoft
3. Oracle
4. Baan
5. JD Edwards
6. System Software Associates

Microsoft?

Pakettiohjelma (esim. ERP)

- Useimmiten on halvinta ja nopeaa ostaa mahdollisimman paljon valmiina
- Valmis ohjelma on valmiiksi testattu, muilla on kokemuksia
- Miten saadaan pakettiohjelma kommunikoimaan muiden järjestelmien kanssa?
- Entä jos voidaan/halutaan sovittaa softasta vain pieni osa omaan käyttöön? -> onko ohjelma modulaarinen?
- Halutaanko sopeuttaa omat prosessit pakettisoftan mukaisiksi?
- Useimmat ohjelmat vaativat kuitenkin muokkausta

Mahdollinen tilanne

Tuotannon
suunnittelu

Varasto-
kirjanpito

Extranet-
palvelut

ERP

- Laskutus
- Reskontrat
- Kirjanpito
- Asiakas-
rekisteri
- Myynti
- Johdon tuki

Palkanlaskenta

Työaika-
seuranta

Internet-
palvelut

Kokonaiskustannukset !!

- Ohjelmiston hinta
- Ohjelmiston käyttöönotto/muokkaus
- Koulutus
- Laitteistosta, tietoliikenteestä yms. aiheutuvat kulut
- Ylläpitokulut
- IT-henkilöstökulut (käyttöönotto, ylläpito, tuki)
- <http://www.sap.com/solutions/smb/>

ERP-projektin mittasuhteet (Meta Group 2003)

- Tutkimuksessa yli 200 "ERP-yritystä"
- Lasketaan kokonaiskustannuksia (TCO) kahden vuoden käytön jälkeen
- Suurissa yrityksissä 1% liikevaihdosta (\$10M -> \$1B)
- Käyttöönotto 20 kk – 27 kk hyötykäyttöön
- Käyttäjää kohti yli 50 000 \$ (aiempi tutkimus)

ERP-projektin ongelmia

(The Top 10 ERP Mistakes, intelligentERP 2003)

3. Trying to create a solution incompatible with the company's culture
2. Treating this as a technical project vs. a change that balances people, process, and technology; not using the power of the new, integrated information
1. Embarking on the journey without a solid, approved business case, including mechanisms to update the business case continuously and to ensure the savings are baked into operational budgets.

http://www.intelligenterp.com/feature/2003/03/0303feat1_2.shtml

EAI (Enterprise Application Integration)

Integroinnin tasot (Linthicum, 1999):

- Data-level EAI
- Application interface-level EAI
- Method-level EAI
- User interface-level EAI

Linthicum, David S., Enterprise Application Integration, Addison-Wesley Pub Co., 1999.

Järjestelmien välinen kommunikointi – one to one



- Täytyy tuntea toinen järjestelmä – ongelma varsinkin, kun useita järjestelmiä
- Ei salli muutoksia
- Synkroninen

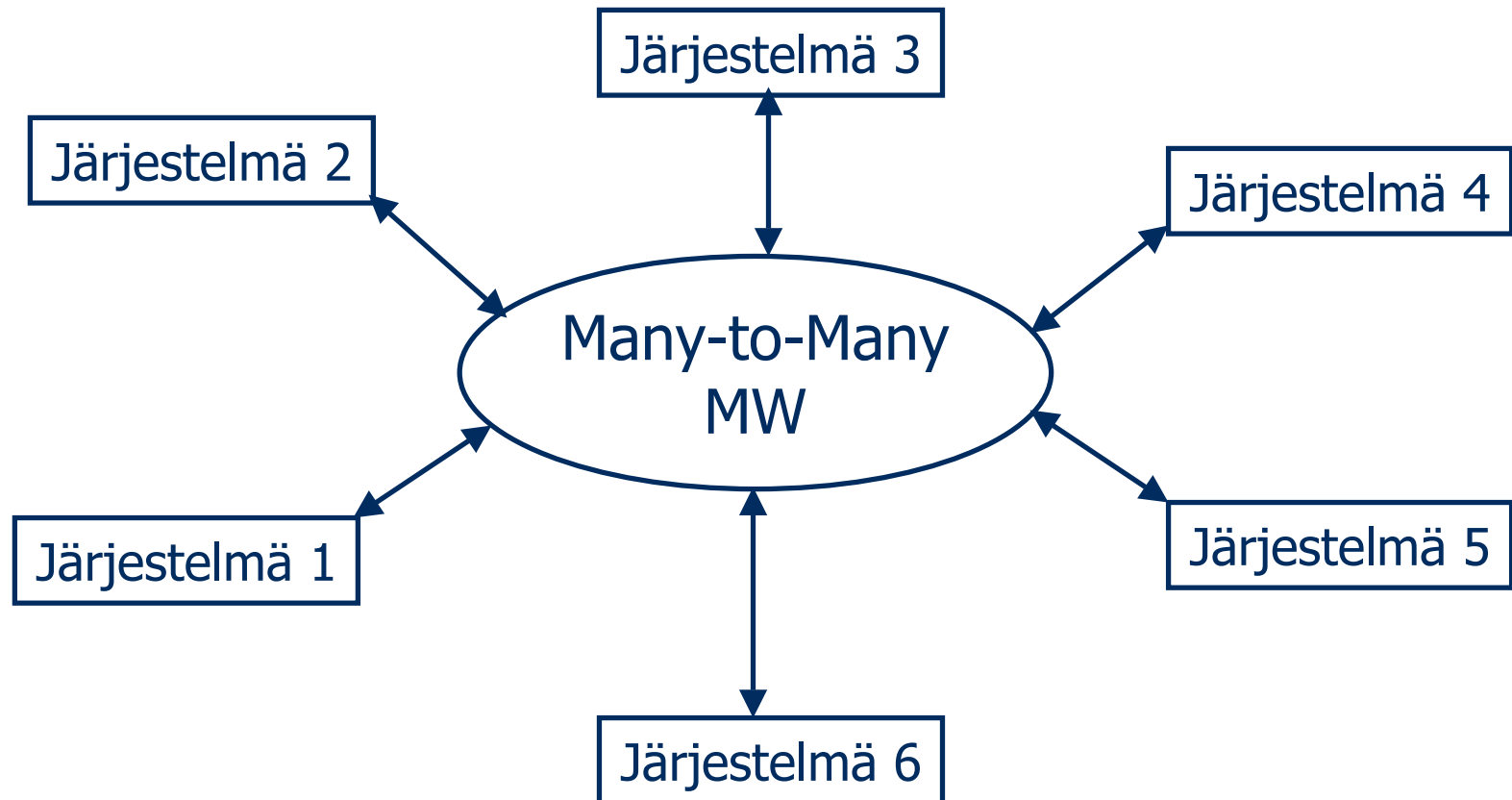
Järjestelmien välinen kommunikointi

– one to one



- Asynkroninen
- Esim. Yksinkertainen jonokäsittelijä
- Tiedon muunnokset – voidaan päivittää toista järjestelmää toisen tietämättä
- Reititys – ei tarvitse tietää toisen sijaintia

Many to Many



Message Broker

- Tiedon muunnokset
- Jonokäsittelijä
- Säännöt tiedon muunnosta ja reititystä varten
- Reitityslogiikka
 - Tietää, kuka vastaa mistäkin palvelusta
 - Voi lähettää saman viestin myös useille
 - Tietää, ketkä kaikki tarvitsevat mitäkin tietoa
 - Missä muodossa kukakin tietoa tarvitsee

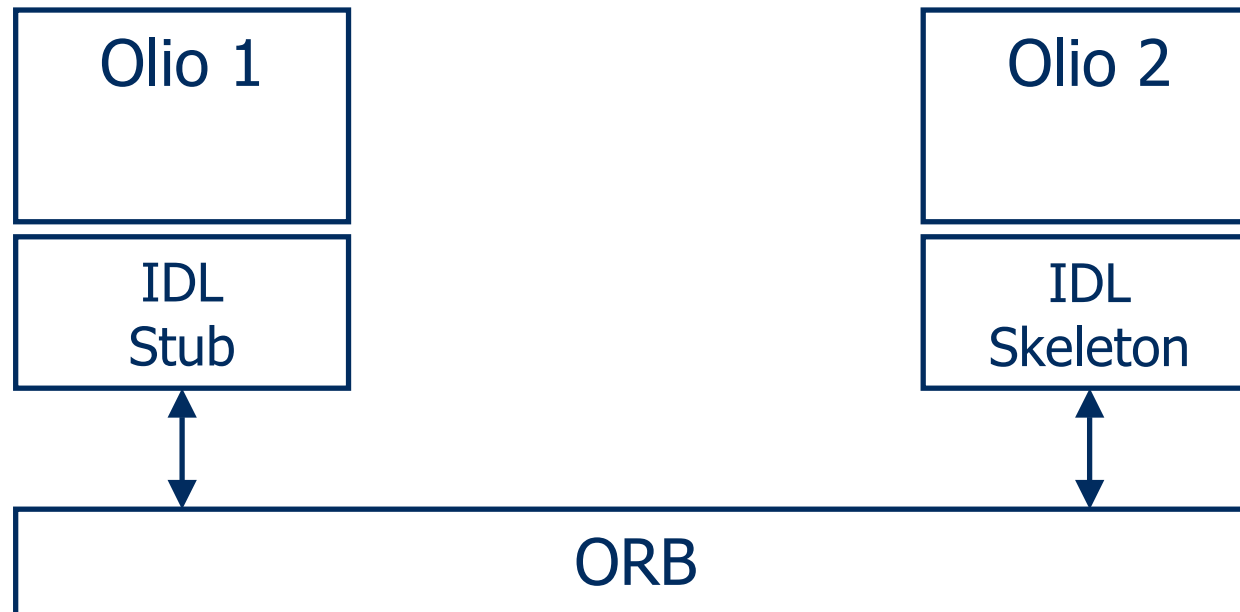
Hajautetut oliot (Distributed objects)

- Integroinnin taso
 - Tiedon jakaminen < toiminnallisuuden jakaminen
- Uusiin järjestelmiin
- Etua, kun on tarve jakaa paljon yhteisiä metodeja
- Corba vs. DCOM
- Periaatteena, että sovelluksen ei tarvitse paikallistaa kutsuttava oliota

CORBA

- Standardi olioiden jakamisen toteutukseen
- OMG (Object Management Group):
 - IBM, Apple, Sun ym.
- Julkaistiin vuonna 1991, mutta ei tuloksia vielä moneen vuoteen sen jälkeen
- Tuki useimmille ohjelmointikielille
- Ongelmia:
 - Suorituskyky
 - Palautumismekanismien puuttuminen
 - Huono skaalautuvuus

CORBA TOTEUTUS



IDL = Interface Definition Language

ORB = Object Request Broker

Kutsu asiakkaan kannalta

- Paikkatuntumaton
- Palvelintuntumaton
- Ohjelmointikieliriippumaton
- Toteutusriippumaton
- Laitearkkitehtuurista riippumaton
- Käyttöjärjestelmästä riippumaton
- Protokollista riippumaton
- Siirtokerroksesta (fyysinen verkkoyhteys) riippumaton

ORB:n toiminta palvelua pyydetessä

- Paikallistaa kohdeolion
- Aktivoi palvelinohjelman, jos se ei ole käynnissä
- Siirtää pyynnön parametrit
- Odottaa prosessin suoritusta
- Palauttaa paluuarvon
- Palauttaa poikkeuksen, jos ei onnistu

.NET

- Microsoftin ratkaisu sovellusten integrointiin
- Tuotteistettu kehitysympäristö, jossa on kymmeniä ohjelmointikieliä
- Ohjelma käännetään välikielelle, kuten Javassa
- Common Language Runtime (CLR) –virtuaalikone
- CLR useille alustoille?
- Web Services ja XML -> toimii myös muilla tekniikoilla toteutettujen järjestelmien kanssa
- Onko .NET tulevaisuus?

Huomioita arkkitehtuurisuunnitteluun

- Arkkitehtuuri avoimeksi
 - Huomioitava mahdollisuus integrointiin
- Käytetään standarditapoja:
 - XML, J2EE, .NET ...
- Valitaan omaan tarkoitukseen parhaiten sopiva toteutustapa