

# *Interfacing Product Data Management System*

Tekijä: **Mats Kuivalainen**  
Työn valvoja: **Timo Korhonen**



## Esitelmän sisältö

- Työn suorituspaikka - Ideal Product Data Oy**
- Käsitteitä**
- Työn tavoitteet**
- Työn tulokset 1/5 – mahdolliset toteutustavat**
- Työn tulokset 2/5 - ohjelmointirajapintojen käyttö**
- Työn tulokset 3/5 - suora tietokantayhteys**
- Työn tulokset 4/5 - siirtotiedostojen käyttö**
- Työn tulokset 5/5 - liitynnän toteutus**
- Johtopäätökset**

## Työn suorituspaikka - Ideal Product Data Oy



### Siemens PLM Softwaren kumppani Suomessa

- Perustettu 1992
- Toimistot Espoossa ja Tampereella
- Suomen suurin PDM-järjestelmien toimittaja
- Työntekijöitä noin kolmekymmentä

[www.ideal.fi](http://www.ideal.fi)

## Käsitteitä

- **Product Data Management (PDM)**

- Suomeksi tuotetiedonhallinta.
- Menetelmä, jolla ohjataan tuotteen kehitystä sen koko elinkaaren ajan.
- PDM-järjestelmä antaa oikean tuotetiedon oikeaan aikaan oikeille henkilöille.

- **Liityntä**

- Kahden tietojärjestelmän välille toteutettu tekninen ratkaisu joka mahdollistaa tiedonsiirron järjestelmien välillä.
- Yksi- tai kaksisuuntainen.
- PDM-järjestelmällä voi olla liityntätarpeita esimerkiksi muihin suunnittelujärjestelmiin ja tuotannonohjausjärjestelmiin.
- Tarvitaan, jotta samaa tietoa ei tarvitse syöttää käsin useaan kertaan.

## Työn tavoitteet

- **Työn tavoitteena oli**
  - Selvittää millaisilla ratkaisuilla liityntä PDM-järjestelmän ja jonkin toisen järjestelmän välillä on mahdollista toteuttaa.
  - Koota tieto liityntöjen tekoon liittyvistä standardeista yhteen.
  - Toteuttaa yksi liityntä käytännössä.

## Työn tulokset 1/5 – mahdolliset toteutustavat

- **Kolme tapaa toteuttaa liityntä**
  - Ohjelmointirajapintojen käyttö.
  - Suora tietokantayhteys.
  - Siirtotiedoston käyttö.

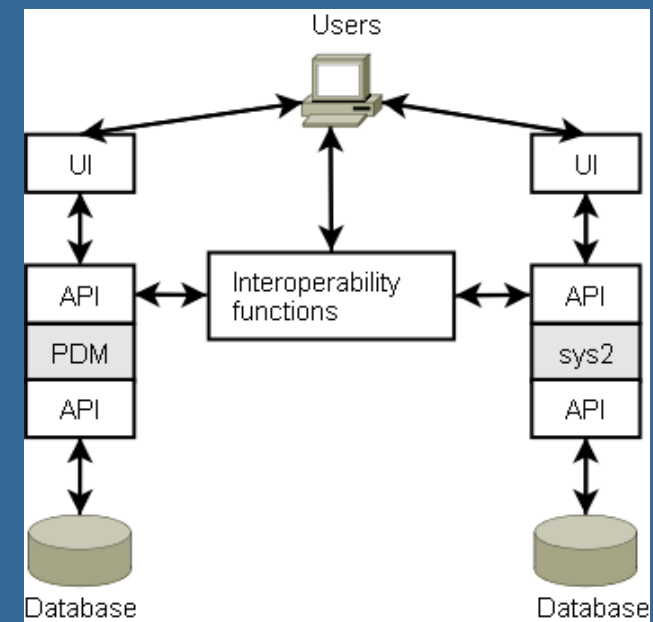
## Työn tulokset 2/5 - ohjelmointirajapintojen käyttö

- **Ohjelmointirajapinta eli API (Application Programming Interface)**

- Tarjoaa mahdollisuuden päästä käsiksi tietojärjestelmässä olevaan dataan esimerkiksi java- tai c-kielellä.

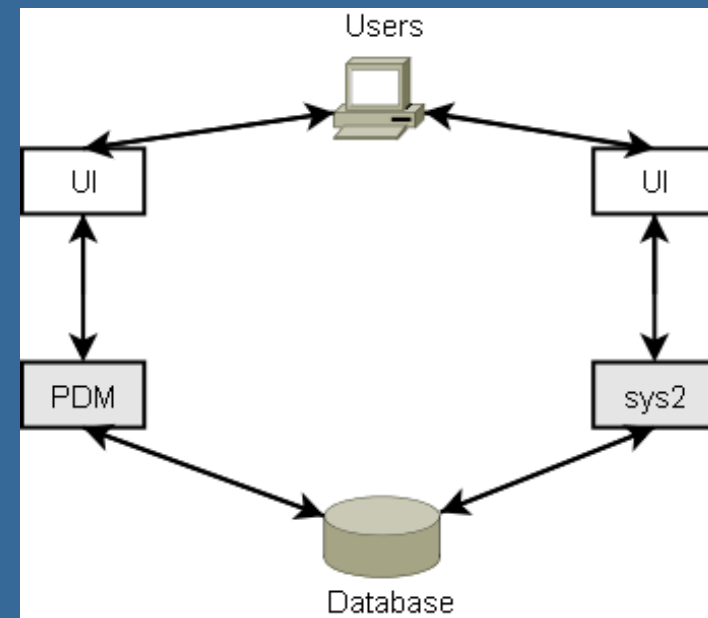
- **Interoperability functions**

- Järjestelmien välissä toimiva sovellus, joka järjestelmien API:en avulla toteuttaa tiedonsiirron.
- Käyttää esim. PDM-järjestelmän API:n funktioita datan lukemiseen ja toisen järjestelmän API:n funktioita datan kirjoittamiseen.



## Työn tulokset 3/5 - suora tietokantayhteys

- PDM-järjestelmä käyttää toisen järjestelmän kanssa samaa tietokantaa: lukee sieltä tietoja tai jopa kirjoittaa sinne.
- Edellyttää hyvää tuntemusta toisen järjestelmän tietomallista.





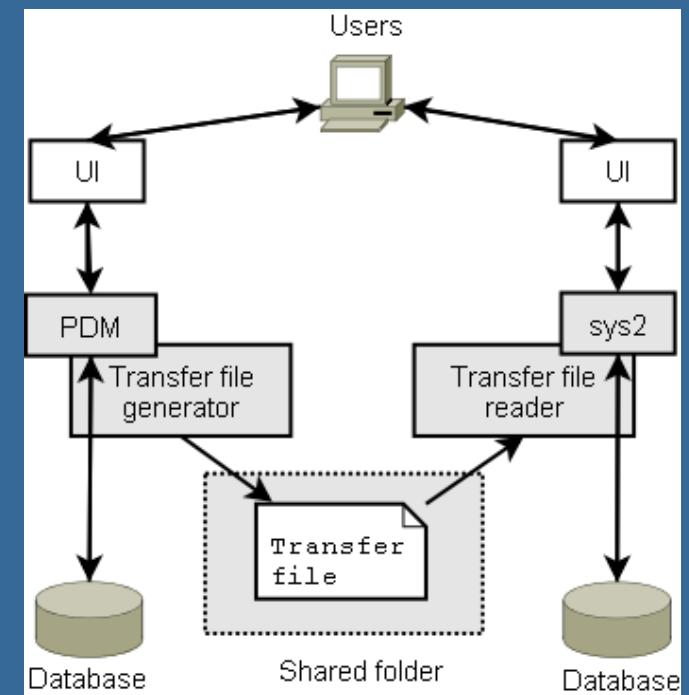
## Työn tulokset 4/5 - siirtotiedostojen käyttö

- Ensimmäinen järjestelmä kirjoittaa datan fyysiseen tiedostoon, josta toinen sitten lukee sen.

- Turvallinen tapa toimia, koska toisen järjestelmän tietokannan sisältöön ei kajota.

- **Standardeja**

- STEP (Standard for the Exchange of Product Data )
- PDX (Product Data eXchange)
- PLM XML (Product Lifecycle Management XML)
- XSLT (eXtensible Stylesheet Language: Transformations)



## Työn tulokset 5/5 - liitynnän toteutus

- **Lähtökohtana kaksi PDM-järjestelmää**
  - Tarve saada siirrettyä dataa vanhasta järjestelmästä uuteen.
- **Toteutus siirtotiedostoilla**
  - Lähettävästä järjestelmästä sai tietoa ulos PDX-formaatissa.
  - Vastaanottava järjestelmä kykeni vastaanottamaan tietoa PLM XML formaatissa.
  - Sekä PDX että PLM XML ovat XML-pohjaisia formaatteja, joten PDX:n muuttaminen PLM XML muotoon onnistui XSLT-tekniikan avulla.
  - Muunnosta varten piti tehdä ns. tyylitiedosto, jossa tarvittavat muunnokset määriteltiin.
  - XSLT-prosessorina eli ohjelmana, joka suoritti muunnokset, käytettiin Xalan-nimistä ohjelmaa.

## Johtopäätökset

- Siirtotiedostojen avulla voidaan toteuttaa toimiva liityntä.
- Siirtotiedostojen käyttöä parempi vaihtoehto aikakriittisissä liitynnöissä olisi todennäköisesti ohjelmointirajapintojen käyttö.
  - Mielenkiintoinen jatko työlle olisikin toteuttaa liityntä ohjelmointirajapintoja käyttäen.
- Siirtotiedostoa käyttämällä vastuukysymykset säilyvät selkeinä.
  - Vastaanottava järjestelmä vastaa aina itse tiedon kirjoittamisesta tietokantaan.
  - Tärkeää varsinkin silloin kun järjestelmillä on eri toimittajat.
- Käyttämällä standardiformaatteja liittymän pitäisi toimia myös ohjelmaversioiden muuttuessa ja liittymä on helposti siirrettävissä muihin samoja standardeja käyttäviin järjestelmiin.