

XML-tietojenkäsittely

Ossi Nykänen

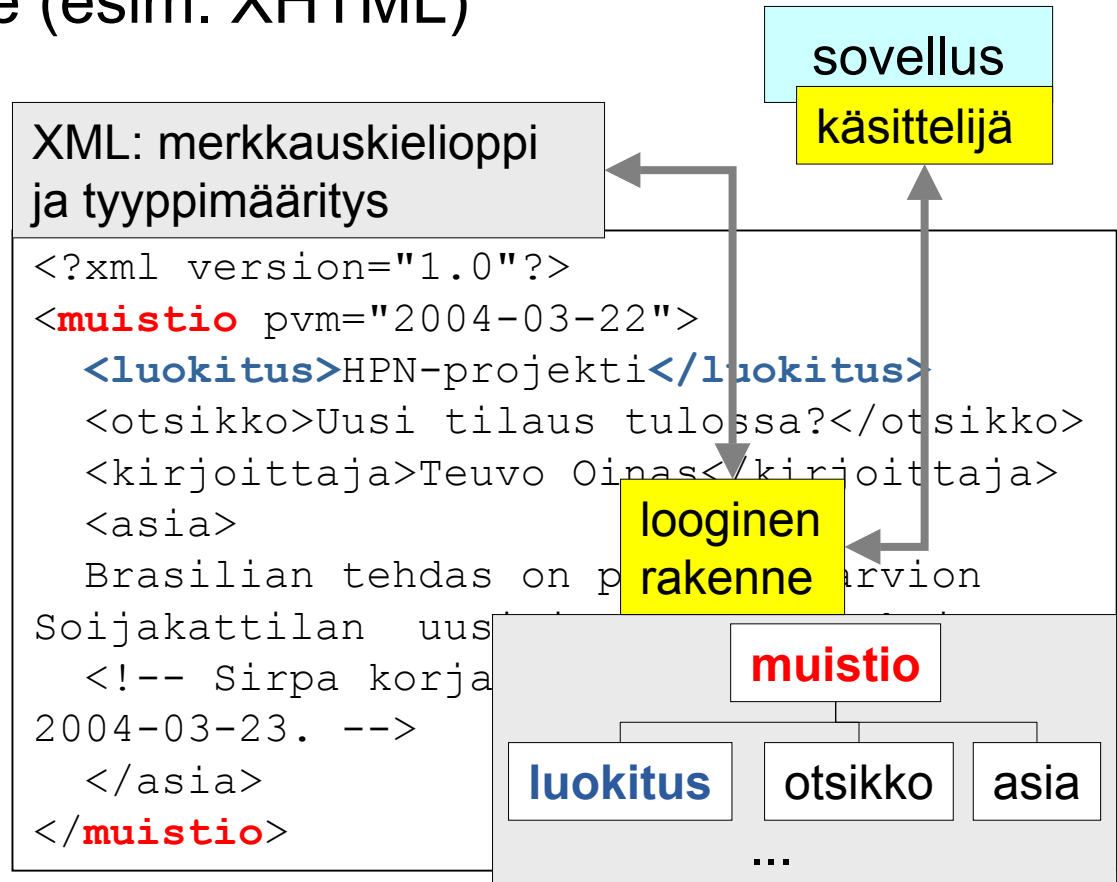
Tampereen teknillinen yliopisto (TTY),
Digitaalisen median instituutti (DMI),
W3C Suomen toimisto



- Hyvin lyhyt versio:
 - XML tarjoaa välineitä tietojenkäsittelyyn eri abstraktiotasoilla. XML-tekniikoiden hyödyt perustuvat yleensä XML-tiedon käsittelyn helppouteen sekä koeteltujen standardisovellusten hyödyntämiseen.
- Pidempi versio, otsikoita:
 - XML-tiedon ohjelmallisen käsittelyn perusteet
 - SAX, DOM
 - XML-prosessorien hyödyntämisestä: XQuery
 - ...XML-middleware
 - kohti datalähtöisiä sovelluksia: RDF Data Access
- Esityksen tavoite: luonnehtia matalan ja korkean XML-tietojenkäsittelyn suhdetta sekä sovellusongelmien jäsentämistä XML:ää hyödyntäen

XML-tiedon ohjelmallisen käsittelyn perusteet (strategia)

- **XML-merkkkaus** → yhtenäinen looginen puurakenne
- **Dokumenttityypit** → tietyn tyyppimäärittelyn mukainen sanasto ja rakenne (esim. XHTML)
- **Standardi-laajennukset** → (periaatteessa) yleiskäyttöiset piirteet (esim. XLink)



- Matalan tason ohjelmointi lähtee liikkeelle XML-dokumenteista **tietorakenteina**
 - ohjelmointia tavallisilla C/C++/Java/... -kehitysvälineillä

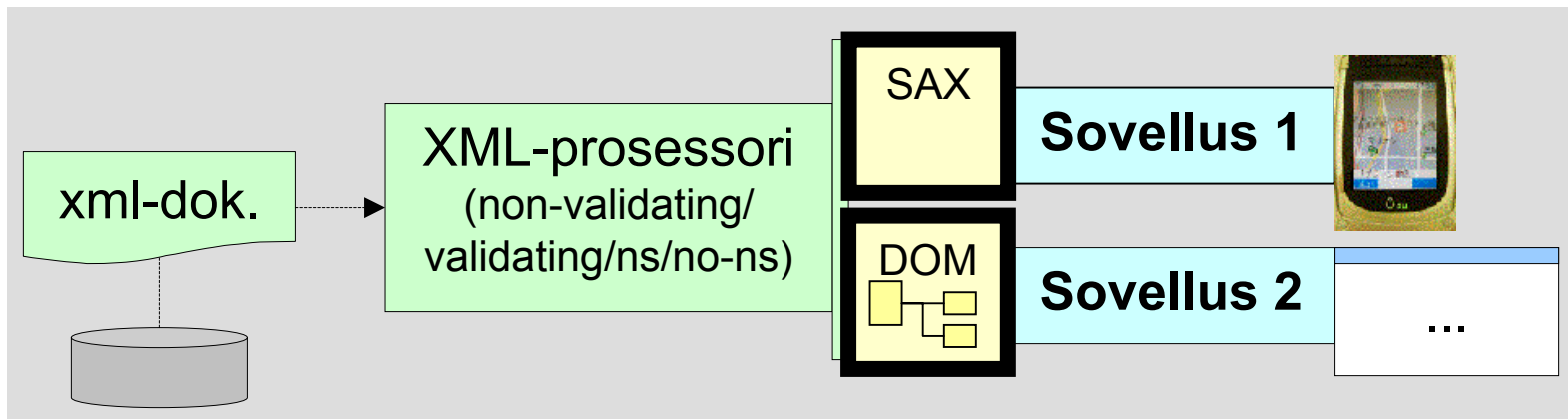
1. Tapahtumapohjainen

ohjelmointi

- SAX-rajapinta (Simple API for XML)

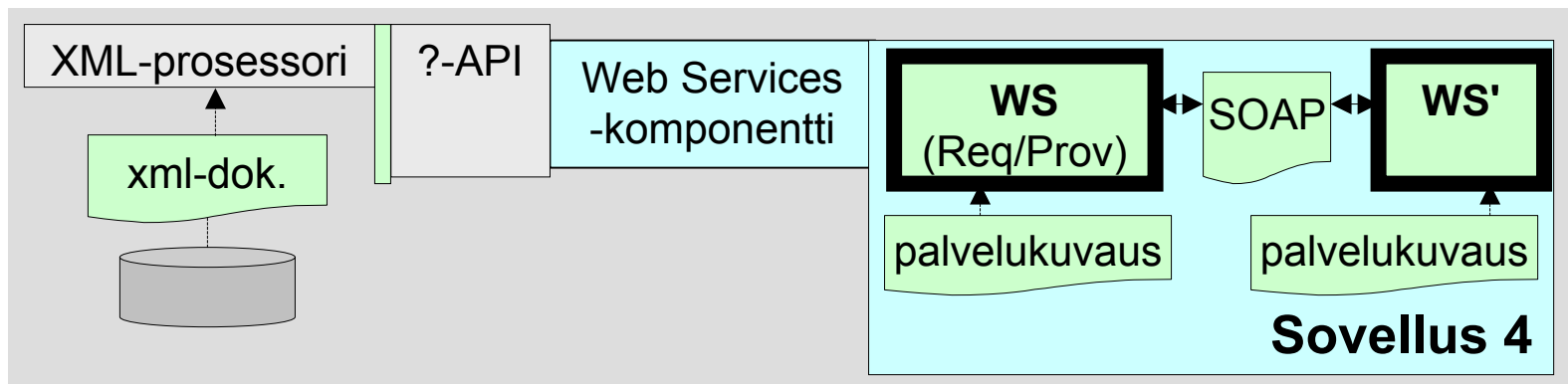
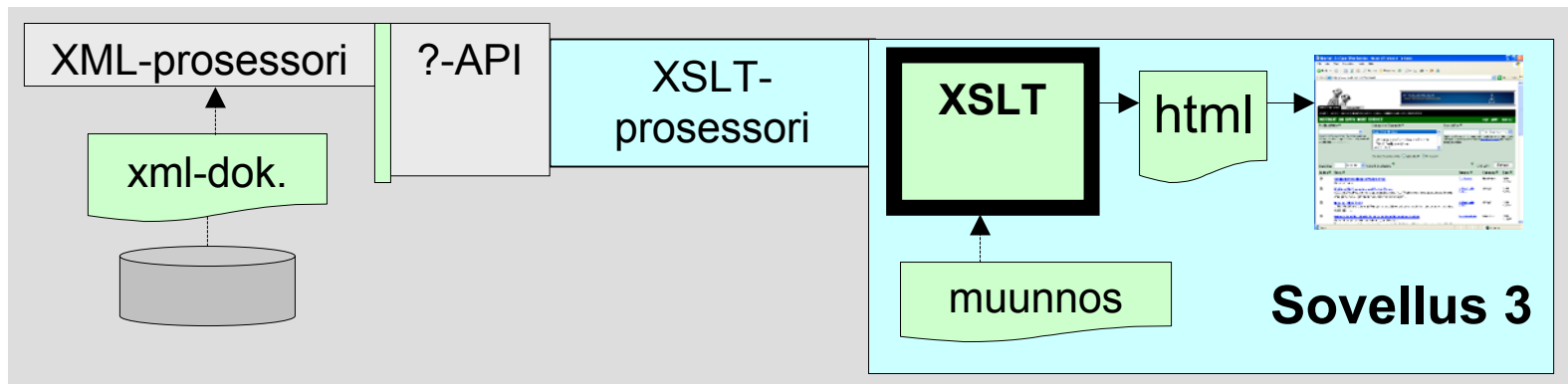
2. Objektipohjainen ohjelmointi

- DOM-rajapinta (Document Object Model)

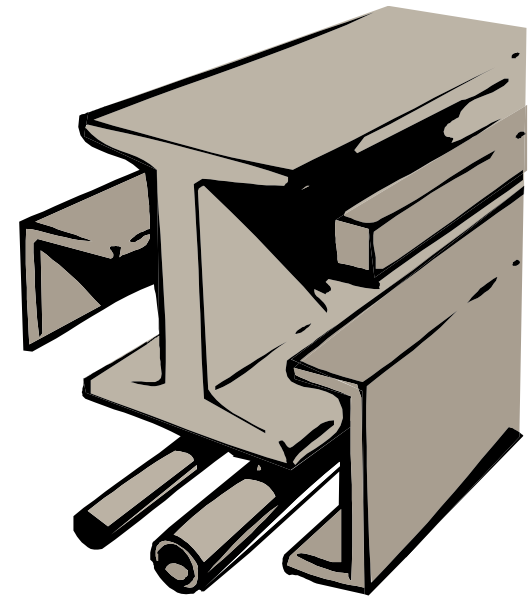


3. Korkeamman tason sovelluskehitys hyödyntää standardisovelluksia (?-proessorit, ?-komponentit)

- sovelluksen kuvaaminen prosessorin/komponentin avulla tai sen syötteenä/vasteena (esim. XQuery, XSLT, Web Services, ...; kyselyt/filtterit/muunnokset/viestit/ ...)



- Täysin räätälöityjen sovellusten tekemisessä on kyse melko "matalan tason" ohjelmoinnista
 - SAX- ja DOM-tyyppisiä rajapintoja (ja jäsentimiä, esim. Xerces) löytyy useimpiin ohjelmointikieliin
- Sovellusten suunnittelussa sisältöjä tarkastellaan yleensä tietomallien tasolla (standardiratkaisut)
- XML-ohjelmointirajapintoja löytyy toki muitakin
 - XML 1.0 määrittelee XML-prosessorin käyttäytymisen, mutta ei rajapintaa, joten sellaisen voi kirjoittaa "kuka tahansa"



Matalan tason XML- ohjelmoinnista: SAX ja DOM

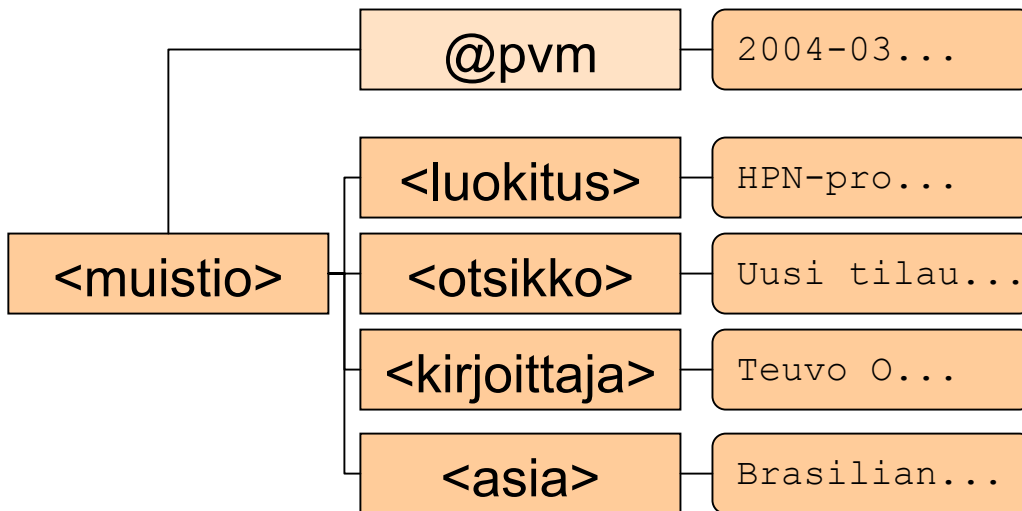
- SAX (1/2) on **tapahtumapohjainen rajapinta** XML-dokumenttien käsittelyyn
 - lähdedokumentin loogiset osat nähdään siinä järjestyksessä kun ne tulevat dokumenttia luettaessa vastaan
 - (v1 ei osaa tallettaa tiedostoja, eikä ymmärrä nimiavaruuksia)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<muistio pvm="2004-03-22">
  <luokitus>HPN-projekti</luokitus>
  <otsikko>Uusi tilaus tulossa?</otsikko>
  <kirjoittaja>Teuvo Oinas</kirjoittaja>
  <asia>
    Brasilian tehdas on pyytänyt arvion
    Soijakattilan uusimisen kustannuksista.
    <!-- Sirpa korjasi kirjoitusvirheet
    2004-03-23. -->
  </asia>
</muistio>
```

SAX-pohjaisen XML-ohjelman suoritus ~

```
startDocument()
startElement("muistio")
startElement("luokitus")
characters("HPN-
projekti")
endElement("luokitus")
startElement("otsikko")
...
endElement("muistio")
endDocument()
```

- DOM (1/2/3) on **objektipohjainen rajapinta** XML-dokumenttien käsittelyyn
 - dokumentti jäsennetään ensin **objektipuuksi**, jossa voidaan liikkua solmusta toiseen kyselyjen ja iteraattoreiden avulla
 - vapaa liikkuminen puurakenteessa (hintana monimutkaisuus ja suurempi muistinkulutus)



DOM-pohjaisen XML-ohjelman idea ~

```
while (  
  
    (child=iterator.nextNode())  
    !=null) {  
    process(child);  
}  
...  
Element liiteElement =  
doc.createElement("liite");  
inode.appendChild(liiteElem  
ent);  
...
```

Abstraktien XML-prosessorien hyödyntämisestä: XQuery

Lausekielinen ohjelmointi?



- Kaikki sovellusohjelmointi ei kuitenkaan ole SAX/DOM-tason C/C++/Java/... -ohjelmointia
- Tyypillisesti toistuvia "korkeamman tason" tehtäviä ovat mm.
 - dokumenttiluokkien väliset muunnokset (esim. muistio→html)
 - kyselyt ja tiedonhaku (vrt. SQL)
- ...joita silmälläpitäen on kehitetty (kehitteillä) esim. standardit
 - **XSL Transformations, XML Query Language, SPARQL**
 - joille löytyy prosessoritoteutuksia (esim. Saxon)
- Käytännön ohjelmointityötä voidaan abstrahoida ja vähentää eri prosessoreja sopivasti yhdistelemällä

- **Kysely** = (pieni) ohjelma tai lauseke, joka palauttaa tietoa XML-dokumentista (tulos tekstiä tai XML-dokumentti)
- XQuery ...
 - perustuu pitkälti XPath-kieleen ideoita mm. SQL-kielestä yms.
 - kysely näkee koko XML-dokumentin (kuten DOM)
 - kysely voi generoida hyvinkin monimutkaisia XML-dokumentteja
- Kyselyillä on oma syntaksinsa (löytyy XML-sarjallistus)
- XQuery sisältää SQL-tyyppiset kutsut erikoistapauksena
 - ns. **FLWOR-lauseke** (for-let-where-order by-return)

```
for $b in doc("raportit-2004.xml")//muistio
where $b/@pvm = "2004-03-22"
return $b/otsikko
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
<muistio pvm="2004-03-22">
  <luokitus>HPN-projekti</luokitus>
  <otsikko>Uusi tilaus tulossa?</otsikko>
  <kirjoittaja>Teuvo Oinas</kirjoittaja>
  <asia>
    Brazilian tehdas on pyytänyt arvion Soijakattilan uusimisen
    kustannuksista.
    <!-- Sirpa korjasi kirjoitusvirheet 2004-03-23. -->
  </asia>
</muistio>
```

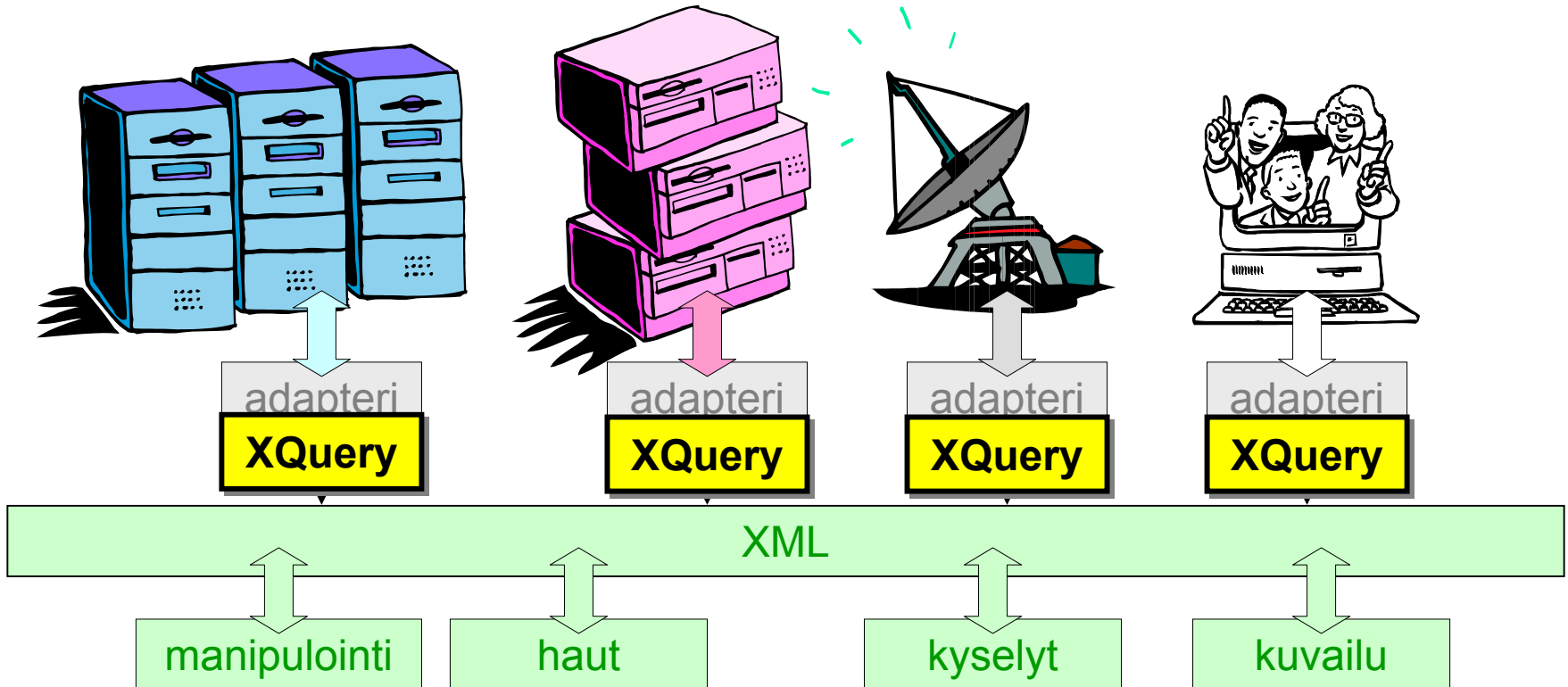
```
<yhteenveto count="{ count(doc('muistio.xml')/muistio//*) }">
{ doc("muistio.xml")//asia }
</yhteenveto>
```

```
<yhteenveto count = "4">
  <asia>Brazilian tehdas on pyytänyt arvion
  Soijakattilan uusimisen kustannuksista.</asia>
</yhteenveto>
```



kysely

Takaisin sovellusintegraatioon...

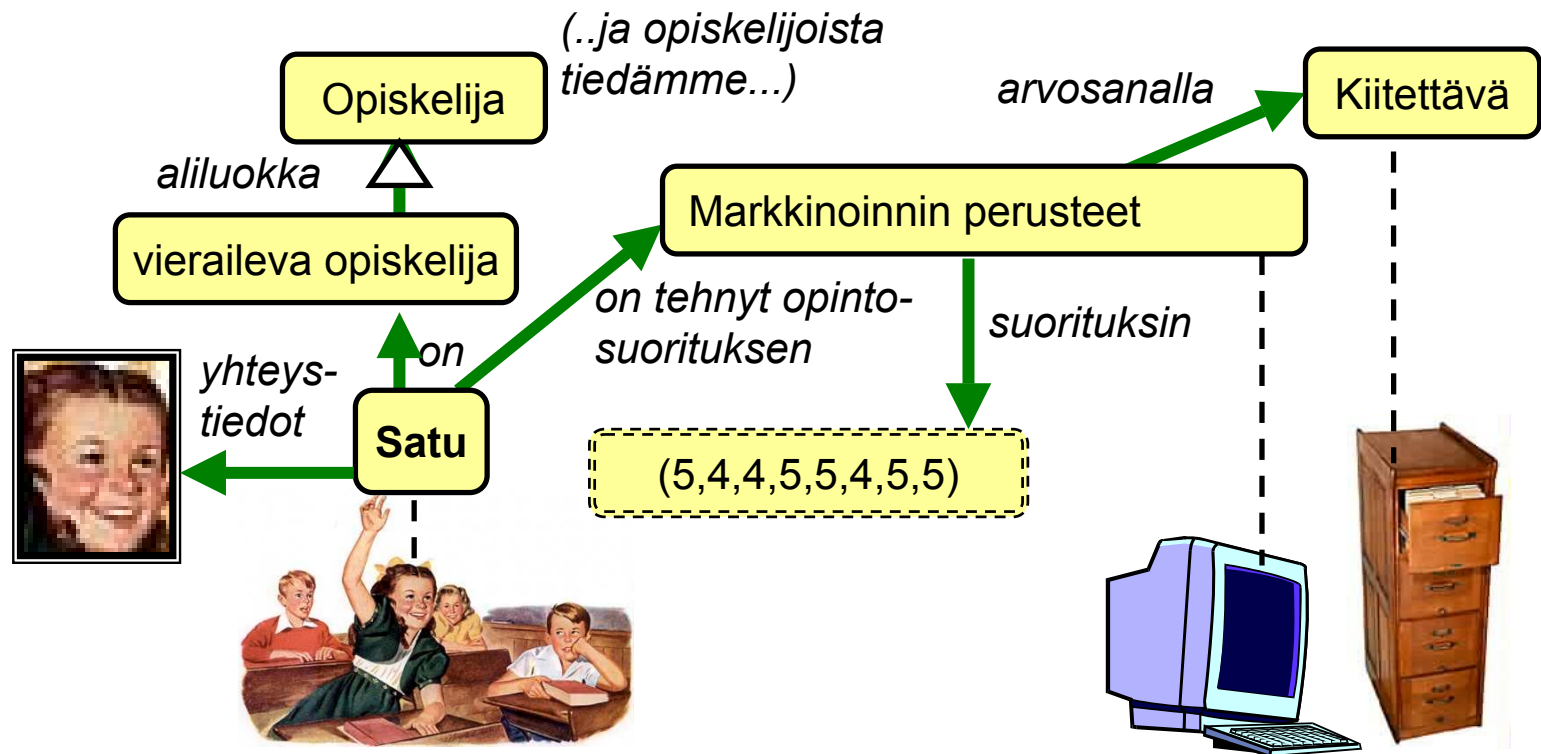


- rajapinnat ja standardiprosessorit tarjoavat tekniikan kääriä mitä tahansa tietoa XML:n avulla (...**virtuaaliset dokumentit**)
- ..."XML everywhere", "XML middleware"

Kohti datalähtöisiä sovelluksia

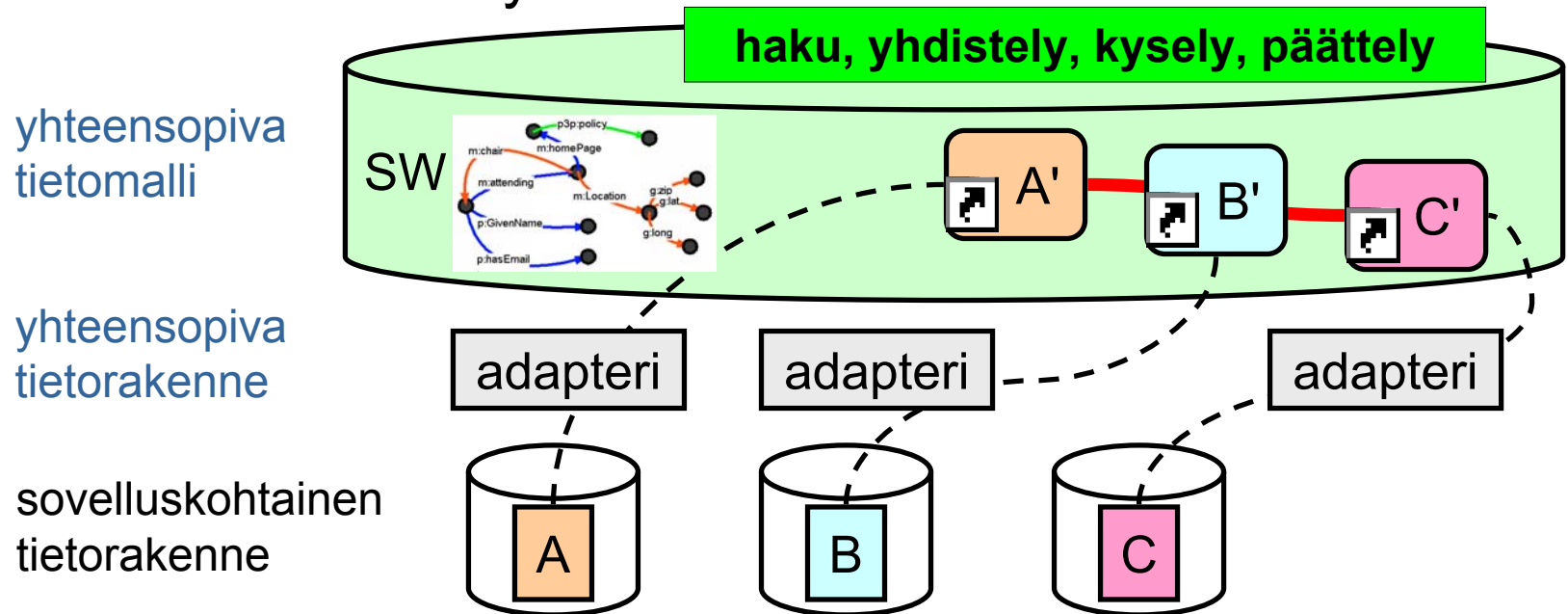
Semanttinen Web?

- Datalähtöisten sovellusten kompastuskivi on yleensä yhteismitallisen tiedon saatavuus
- **Semanttinen Web** tarjoaa perustekniikkaa kuvailevan (meta)tiedon mallintamiseen, julkaisuun ja käsittelyyn



Mistä SW:n (meta)tieto on peräisin?

- Kärjistetyksi, SW voidaan nähdä **universaalin tietokannan** määrittelynä



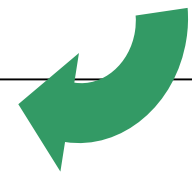
- Tarkoitus ei kuitenkaan ole "siirtää tiedostoja" Semanttiseen Webiin, vaan "kuvata tietoja" (ts. SW ei ole fyysinen vaan **looginen rakenne**)

- "RDQL on Semanttisen Webin SQL"

RDF-lähde

```

SELECT      ?resource, ?familyName
WHERE
  (?resource, <info:age>, ?age) ,
  (?resource, <vCard:N>, ?y) ,
  (?y, <vCard:Family>, ?familyName)
AND
  ?age >= 24
USING
  info FOR <http://somewhere/peopleInfo#> ,
  vCard FOR <http://www.w3.org/2001/vcard-rdf/3.0#>
  
```



?resource	?familyName
http://www.u1...	Romppainen
http://www.u2...	Ala-Harja
http://www.u3...	Kaarto
...	...



sovellus

Lopuksi

- XML luo yhteensopivan perustan ja menetelmiä mitä moninaisemmille sovelluksille
- XML sinänsä ei tietenkään ratkaise ongelmia mutta tarjoaa hyvät puitteet (teknisten suunnittelu-)ratkaisujen löytämiseen
- Menestyvä soveltaja
 - tuntee sovelluksensa
 - osaa perustekniikat (nyt XML) ja
 - tuntee standardiratkaisut ja osaa hyödyntää niitä tarvittaessa
- Kysymyksiä, kommentteja?



- SAX ja DOM
 - <http://www.saxproject.org/>
 - <http://www.w3.org/DOM/>
- XQuery, XSL ja WS
 - <http://www.w3c.org/XML/Query>
 - <http://www.w3c.org/Style/XSL/>
 - <http://www.w3c.org/2002/ws/Activity>
- Semanttinen Web ja RDF Data Access (SPARQL)
 - <http://www.w3c.org/2001/sw/>
 - <http://www.w3.org/2001/sw/DataAccess/>
- + etsi termejä : www.google.fi, www.xml.com, jne