

Avoimen datan julkaisutekniikat

Toni Ylärinne
toni.ylarinne@sovelto.fi



Tämän teos teoksen käyttöoikeutta koskee [Creative Commons Nimeä 1.0
Suomi -lisenssi](https://creativecommons.org/licenses/by/1.0/).

Lähdeviittaus: Helsinki Region Infoshare - Avoimen datan julkaisutekniikat
koulutusmateriaali 2013, Sovelto Oyj.

Agenda

- Johdanto avoimeen dataan, Hami Kekkonen
- 9.30-10.30 Eräajotiedostot, Toni Ylärinne
- 10.30 Case: Eräajotiedostoavaus, Helsinki 20150 –kyselyn vastaukset, Maija Mattila
- 11.15-12.15 Lounas
- 12.15 Dataformaatit ja REST-palvelut, Toni Ylärinne
- 13.15 Case: rajapinta-avaus, Espoo.fi-verkkopalvelun avoin rajapinta, Virpi Pakkala
- 14.15 Avoimen datan hyödyntäminen, Hami Kekkonen
- 15.30 Koulutus päättyy

Johdanto avoimeen dataan

Avoimen datan hyödyt

- Parantaa hallinnon sisäistä tehokkuutta
- Luo uutta liikentoimintaa ja innovaatioita
- Hallinnon läpinäkyvyys ja demokratia paranee

Avoin data

- Avoin data on trendi
 - Tietovarastoja avataan
 - Sovelluskehittäjät tekevät sovelluksia, jotka hyödyntävät avointa dataa
 - Data journalismi on syntynyt
 - Päättäjät ovat motivoituneita avoimen datan edistämiseen

Millaista avoin data on?

- Laillisesti uudelleenkäytettävää
 - Ei sisällä informaatiota, joka on luottamuksellista
 - Käyttöehdot sallivat uudelleenkäytön
- Teknisesti avointa
 - Ohjelmointitekniikka ja alustariippumattomuus
 - Sekä formaatti, että palvelurajapinta
- Maksutonta
- Löydettävää
 - Datakatalogit
- Ymmärrettävää
 - Merkitys ja rakenne on kuvattua

Avoimen datan lähteitä Suomessa

- www.hri.fi
- data.suomi.fi
- fi.thedatahub.org
- www.apps4finland.fi/fi/datakatalogi

Suomen valtio aktivoitunut datan avaamisessa:

- Valtion periaatepäättös julkisten tiedovarantojen avaamisesta
- Valtion avoimen datan käyttöehdot muotoutumassa
- Maanmittauslaitoksen maastotietojen avaaminen 1.5.2012
- Tilastokeskuksen datan avaukset – www.stat.fi
- LVM:n hallinnonalan avauksien selvitys
- Ilmatieteenlaitos

Avoimen datan julkaisuprosessi



<http://www.hri.fi/fi/avaa-dataa/>

Datan julkaisukelpoisuus

- Suuri osa julkishallinnon datasta julkisuuslainsäädännön perusteella julkista.
- Selvitettävää on, ettei tekijänoikeudet rajoita aineiston julkaisemista
- Eikä data loukkaa yksityisyyden suojaa.

Dataformaatit

- PDF, Word ja HTML ovat dokumenttiformaatteja, joissa kuvataan ulkoasua
 - Eivät kuvaa dataa, ohjelmallinen käsittely ja parsinta vaikea/hyvin vaikeaa.
- Hyvä dataformaatti pitää sisällään datakuvauksen, ei ulkoasua
 - Käsittely on ohjelmointikieli- , alusta- ja arkkitehtuuririippumaton
- Käytettyjä dataformaatteja
 - CSV (Comma separated values)
 - XML (eXtensible Markup Language)
 - JSON (JavaScript Object Notation)
 - Excel
 - PC-Axis (Avoin tietokantaformaatti)

Datan julkaisu

- Data voidaan generoida eräajotyyppisesti
 - Julkaisu esimerkiksi Web- tai FTP-palvelimelle
 - Data luonteeltaan staattista
- Data voi olla haettavissa myös palvelusta
 - Esimerkiksi REST-rajapinnan kautta
 - Reaaliaikaista dataa
 - Haettavissa pienillä otoksilla käyttäjää kiinnostava data esimerkiksi mobiililaitteeseen
- Datan käytettävyyttä tyypillisesti lisää se, että sama data on useammassa formaatissa ulkona, sekä ehkä kokonaisen tietomassana, että palvelun kautta pyydettäessä

Käyttöehdot

- Käyttöehtojen pitää sallia datan yksityisen ja kauppallisen uudelleen käytön
- Käyttöehdot pitää olla myös selkeästi esillä
- HRI – tietoaineistojen lisenssi – nimeä
 - <http://www.hri.fi/lisenssit/hri-nimea/>
- Creative Commons –lisenssi

HRI – tietoaaineistojen lisenssi - nimeä

1. Tällä lisenssillä lisenssinantaja (lisensoidun aineiston tekijän- tai lähioikeuksien haltija) antaa lisenssinsaajalle (lisensoitua aineistoa tämän lisenssin mukaisesti käyttävä) kohdissa 2 ja 3 mainituin ehdoin maailmanlaajuisen, maksuttoman, ei-yksinomaisen ja pysyvän luvan kopioida, levittää, muokata, yhdistellä sekä muutoin käyttää lisensoitua aineistoa. Lisensoitua aineistoa voi käyttää sekä ei-kaupallisiin että kaupallisiin tarkoituksiin.
2. Lisensoidun aineiston alkuperäiset tekijänoikeustiedot on ilmoitettava lisenssinantajan ilmoittamalla tavalla. Lisenssinantajan pyynnöstä tämä viittaus on poistettava.
3. Lisensoidun aineiston tekijänoikeustietoja ei saa ilmoittaa siten, että ilmoitus viittaisi lisensoidun aineiston tekijän millään tavoin tukevan lisensoidun aineiston käyttäjää tai aineiston käyttötapaa.

CC License Nimeä 3.0 Muokkaamaton

- **Sinulla on vapaus:**
 - **Jakaa** — kopioida, levittää, näyttää ja esittää teosta
 - **Remiksata** — valmistaa muutettuja teoksia
 - käyttää teosta kaupallisiin tarkoituksiin
- Seuraavilla ehdoilla
 - **Nimeä** — Teoksen tekijä on ilmoitettava siten kuin tekijä tai teoksen lisensoija on sen määrännyt (mutta ei siten että ilmoitus viittaisi lissenssiantajan tukevan lissenssinsaaajaa tai Teoksen käyttötapaa).
- <http://creativecommons.org/licenses/by/3.0/deed.fi>

Eräajotiedostot

Eräajotiedostot

- Avoimen datan julkaisussa yleensä helpoin ja suoraviivaisin tapa on generoida tietojärjestelmästä eräajotiedosto
- Dataformaatti vaihtoehtoja on useita
- Hyvä vaihtoehto, kun data on luonteeltaan staattista. Esimerkiksi
 - Tilastodatat joltakin määrätyltä ajalta
 - Budjettidata
- Eräajotiedostot mahdollistaa, että datan hyödyntäjä saa kaiken datan helposti kerralla käyttöönsä, jota hän voi vapaasti sitten käsitellä omien tarpeiden mukaisesti
- Datan käyttäjälle on usein hyödyllistä, että vaikka data olisi REST-rajapinnan kautta noudettavissa, se olisi helposti ladattavissa myös yhtenä tiedostona
- Helpoin julkaisu paikka datalle on luonnollisesti organisaation Web-palvelin tai esimerkiksi FTP-palvelin

Eräajotiedostot

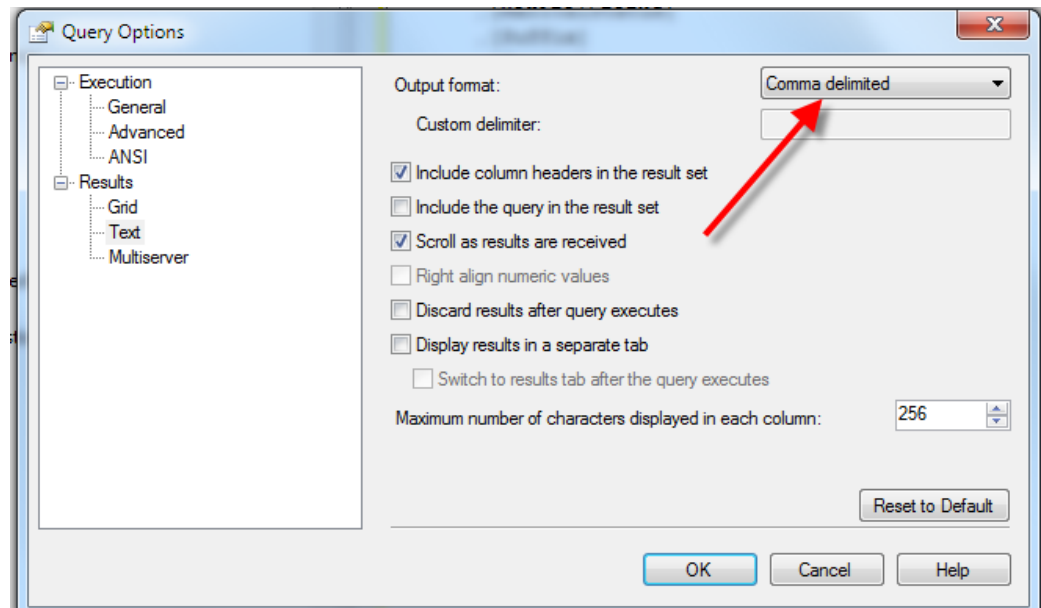
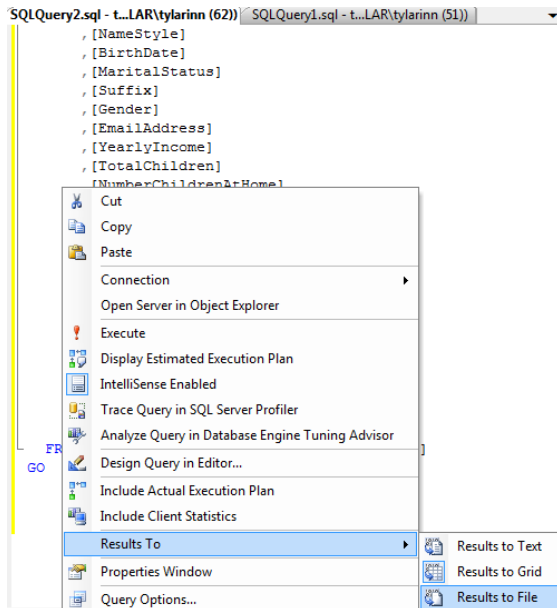
- Datan luonteesta riippuen eräajotiedostot ovat generoitavissa tarpeen mukaan tai vaikka ajastetusti aamuöisin
- Käytännössä datan hyödyntäjä tyypillisesti lataa tiedoston itselleen ja käsittelee sitä sen jälkeen omassa ympäristössään
 - Eli data voi hyvin olla pakattu esimerkiksi zip:iksi. XML, CSV ja JSON tiedostot pakkautuvat erittäin tehokkaasti!
 - Datan julkaisu ei aiheuta yleenä paljon kuormaa julkaisijan Web-palvelimelle
- Jos datan hyödyntäjä haluaa käyttää dataa hyödyksi esimerkiksi tekemässään mobiilisovelluksessa, hän joutuu mahdollisesti toteuttamaan oman palvelun datan julkaisulle

Eräajotiedostojen luontivaihtoehtoja

- Useat tietokannat pitävät sisällään valmiita toimintoja joilla eräajotiedostot ovat luotavissa
 - Formaatti voi olla CSV:ää
 - tai esimerkiksi XML:ää
- Yksi vaihtoehto on tuoda data Exceeliin muotoilla vielä siellä ja exportata ulos csv:änä
- Oman esimerkiksi komentoriviltä tai ajastetusti ajettavan sovelluksen tekeminen ei ole myöskään kehittäjältä haastava urakka

CSV:n exporttaus SQL Serveristä

- Komentoriviltä esimerkiksi
 - `sqlcmd -S <palvelin> -d <Kanta> -E -o "data.csv" -Q "select * from taulu" -W -w 900 -s ",",`
 - `-W` poistaa välilyönnit sarakkeista, `-s` määrittää erottimen, `-w` määrittää rivin maksimi pituuden, `-h-1` poistaisi headerin
 - <http://msdn.microsoft.com/en-us/library/ms162773.aspx>
- Suoraan Management Studiosta



XML:n generoiminen tietokannasta

- Yksinkertaisimmillaan XML-dataa voi generoida tietokannasta SQL-kielen merkkijono-operaatioilla
- Syntaksin virheettömyys jää tällöin käyttäjän vastuulle
- Esimerkiksi SQL-komento:

```
SELECT '<asiakas nro="' || asnro ||  
'"><etunimi>' || etunimi ||  
'</etunimi><sukunimi>' || sukunimi ||  
'</sukunimi></asiakas>' FROM asiakas  
WHERE asiakasnro = 123;
```

tuottaa tuloksena seuraava XML-dokumentin:

```
<asiakas nro="123">  
<etunimi>Ville</etunimi>  
<sukunimi>Virtanen</sukunimi>  
</asiakas>
```

Komentolaajennukset

- SQL/XML-standardi määrittelee joukon funktioita, joiden avulla relaattiorakenteissa oleva data voidaan kuvata XML-rakenteiksi
 - mm. Oracle 10g tukee näitä funktioita
- XML-datan generointiin voi myös olla tuotekohtaisia syntaksilaajennuksia, joiden avulla voidaan määritellä SQL-kielen SELECT-lauseen tulostulosjoukko palautettavaksi XML-rakenteen sisään pakattuna
 - Oracle 10g: DBURITYPE
 - SQL Server 2000/2005: FOR XML
 - helpompia käyttää kuin SQL/XML-standardirakenteet, mutta eivät keskenään yhteensopivia

Esimerkki: Oracle 10g SQL/XML

- Oracle tukee SQL/XML-standardin mukaisia rakenteita
- Esimerkki:

```
SELECT
XMLElement("asiakas",
  XMLAgg(
    XMLElement("row",
      XMLForest(nimi, osoite)))
FROM testi;
```
- Suurin osa koodista tarvitaan juurielementin lisäämiseen tulokseen!

Esimerkki: SQL Server 2005 FOR XML

- SQL Serverissä on SELECT-komennossa FOR XML -laajennus:
SELECT * FROM asiakas FOR XML RAW;
 - oletuksena sarakkeiden arvot kuvataan attribuuteiksi, elementtiesitys saadaan ELEMENTS-avainsanalla
 - XML-juurielementti lisätään ROOT-määreellä
- Monimutkaisemmat rakenteet (mm. sisäkkäiset elementit, attribuuttien ja elementtien yhdistelmät) voidaan määritellä EXPLICIT- tai PATH-määreillä
 - myös AUTO-määre mahdollinen

Dataformaatit

Excel

- Monipuolinen hyvin monelle tuttu taulukko-ohjelmisto, johon pystyy varsin hyvin tuomaan dataa eri järjestelmistä
- Data on helposti muotoiltavissa
- Visualisointi onnistuu samalla tuotteella
- Office Open XML (OOXML/OpenXML) on Microsoftin kehittämä ja ECMA:n standardoima periaatteessa avoin formaatti
- Käytännössä Excel-dokumentteja vasten on varsin hankala tehdä sovelluskehitystä.
- Toisaalta Excel antaa mahdollisuuden exportata ulos esimerkiksi CSV:tä (Comma-Separated-Values)

CSV – Comma-separated values

- Hyvin vanha plain text –formaatti, jota on käytetty pitkään tiedonsiirtoon järjestelmien välillä

Anna,Malli,anna@malli.fi

Mikko,Malli,mikko@malli.fi

- Formaatti on niin ihmisten kuin sovellusten luettavissa ja tuotettavissa missä vain.
- Merkistö encoding voi tuottaa joskus ongelmia
- Ei ole standardia tapaa määrittellä rakennetta eikä validoida

XML

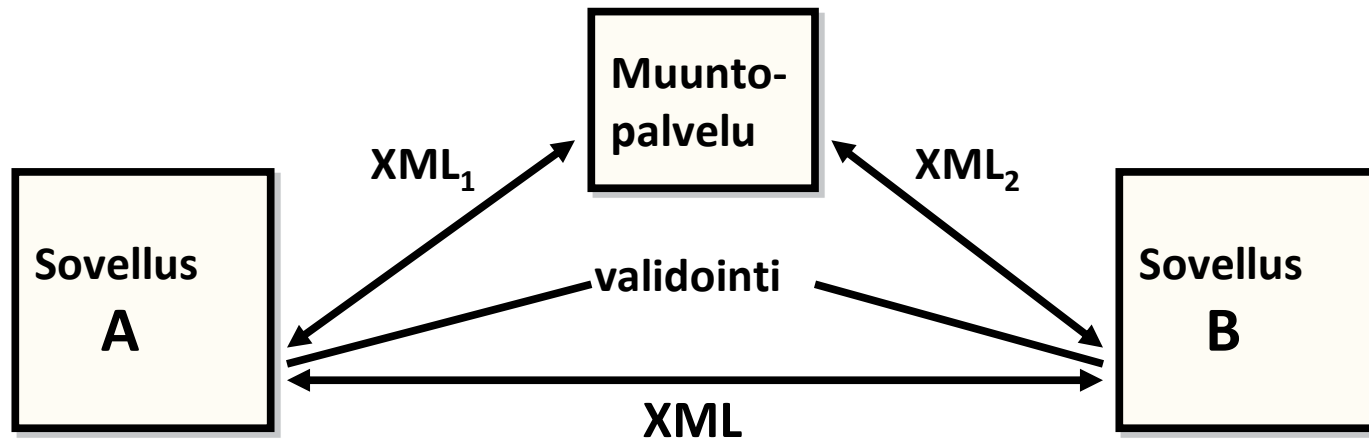
Mitä on XML?

- XML on tekniikka, jolla kuvataan datan *sisältöä*, ei ulkoasua
- Miniesimerkki:

```
<kirja>  
  <tekija>Julius Caesar</tekija>  
  <nimi>Gallian sota</nimi>  
  <kustantaja>Karisto</kustantaja>  
</kirja>
```

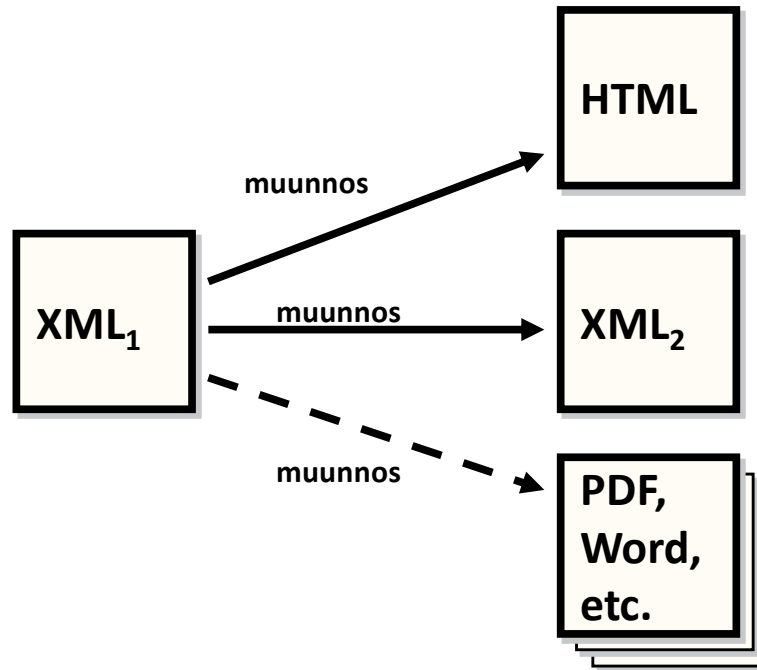
Mihin XML:ää käytetään?

- XML toimii yleisenä dataformaattina myös tiedon siirrossa sovellusten välillä
 - Käyttöjärjestelmäriippumaton formaatti
 - Unicode-tuki
 - Mukana lisäpiirteitä, kuten eheyden varmistus (validointi) tai muunnokset



Mihin XML:ää käytetään?

- XML toimii yleisenä dataformaattina, koska eri esitysmuotojen (esim. HTML tai WML) luominen on tehokasta



Miksi XML?

- Omien dataformaattien siirrettävyys huono
- Yksinkertainen
- Avoin standardi
- Käsittelyn tekniikat (sekä kuvaustekniikat kuten XML Schema tai XSLT että ohjelmointitekniikat kuten DOM) ovat pitkälti kieli- ja alustariippumattomia
- Formaattina ihmisen luettavissa
- Soveltuu "mihin tahansa" tilanteeseen
 - Vertaa: TCP/IP

XML nyt?

- XML on vakiinnuttanut asemansa
- Tärkeimmät XML-sisällön käsittelyyn liittyvät tekniikat ovat valmiita
- Ohjelmointitekniikoista ja sovelluksista löytyy usein jonkinlainen tuki XML:n hyödyntämiselle
- Sanastoja (*vocabulary*) eli eri tarkoituksiin sovittuja elementtijoukkoja on suuri määrä

XML Schema

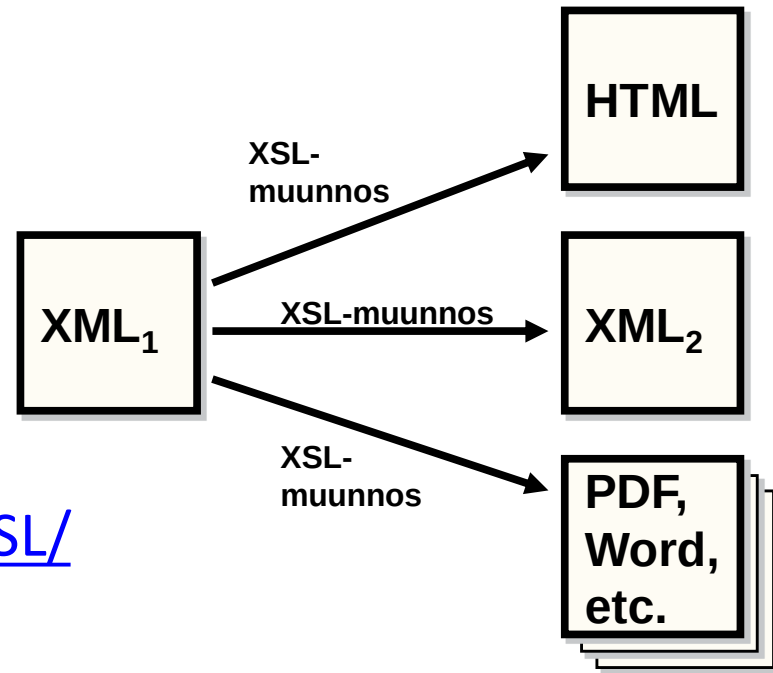
- XML Schema on standardi XML 1.0:n DTD-määrittelyä tarkemman rakenteen sekä tietotyyppien määrittämiseksi XML-tiedostoille
 - Tekniikasta käytetään joskus myös nimeä W3C Schema
- Schema-tiedostojen syntaksina on XML
- Nykyinen kaksiosainen standardi (structures/datatypes) valmistui 2.5.2001
- XML Schema on varsin monimutkainen tekniikka, jossa yksi asia voidaan toteuttaa useammalla tavalla. Perehtyminen hyviin käytäntöihin on suositeltavaa
- <http://www.w3.org/standards/xml/schema>

Käyttökohteita

- Käytetyn XML-formaatin dokumentoiminen
- Datan validoiminen
- Schema-määrittelyksiä voidaan käyttää pohjana suunniteltaessa formaattien välistä XSLT-muunnosta
- Yleisestikin sovellukset, jotka tarvitsevat mahdollisuuden kuvata XML-dokumentin rakenteen yksiselitteisellä tavalla
 - Esimerkki: XML-datan oliomainen käsittely

XSL

- Extensible Stylesheet Language (XSL) on tapa muuntaa XML-dokumentti toiseen XML-formaattiin
 - Käytetään usein sisällön muuntamiseksi HTML-formaattiin
 - Tärkeää myös muissa formaattimuunnoksissa (kohteena toinen XML-sanasto tai muu esitysformaatti)
- <http://www.w3.org/Style/XSL/>



XML-sanastot

Valmiit XML-sanastot

- Maailmassa on julkaistu tuhansia XML-sanastoja erilaisiin tarkoituksiin
- Jotta voitaisiin toteuttaa geneerisiä avoimen datan sovelluksia tiettyä tarkoitusta varten yhteisten sanastojen käyttö olisi hyvin olennaista
- Samoihin tarkoituksiin on valittevasti yleensä useita erilaisia sanastoja
- Parhaimman vaihtoehdon valitseminen voi olla joskus hankalaa
- Sanastojen yleistyminen voi
 - johtua alan jostain merkittävästä standardoimisorganisaatiosta
 - olla maantieteellistä
 - olla tietyn merkittävän toimijan markkinointikyvystä kiinni

XML-dokumentin suunnittelu

- Tutki ensimmäiseksi, onko maailmassa jo julkaistu valmiita XML-sanastoja tarvitsemaasi käyttöön
- Käytä mahdollisuuden mukaan valmiita standardeja XML-sanastoja osa omaa sanastoasi
 - Esim. paikkatieto KML tai GML
- Suunnittelu kannattaa aloittaa tekemällä pohjaksi dokumentti, joka täyttää kyseisen hetken vaatimukset
- Jos XML-dokumenttia tullaan käsittelemään muunnoksissa (XSLT), tee XML-dokumentti jo tässä vaiheessa "valmiiksi" siten, että sen pohjalta ei jouduta tekemään esim. laskentaa
 - Yleensä liian kuvaava (*verbose*) dokumentti on tällöin parempi kuin liian vähän tietoa sisältävä
- Vältä saman elementti- tai attribuuttinimen käyttöä useassa merkityksessä

XML-dokumentin suunnittelu

- Elementtien ohjelmallinen käsittely on joissakin tilanteissa helpompaa kuin attribuuttien käsittely
 - Joskin hyvin laajoissa dokumenteissa voidaan haluta säästää tilaa attribuutteja käyttämällä
- Käytä nimiavaruuksia
 - Uuden sisällön lisääminen helpottuu
- Ole johdonmukainen nimeämisessä
 - Vain yksi kieli
 - Pien- ja/tai suuraakkosten käyttö sovitulla tavalla (vaihtoehtoja on paljon, erityisesti yhdyssanojen ja lyhenteiden kuvaamisen osalta)
- Lopuksi on syytä dokumentoida tehdyt valinnat ja päätökset esimerkiksi Schema- tai DTD-tekniikoiden avulla

Keyhole Markup Language - KML

- Keyhole Inc:n ja Googlen kehittämä kevyt XML-sanasto paikkatiedon kuvaamiseen
- Googlen ja yksinkertaisuutensa takia paljon käytetty
- <https://developers.google.com/kml/documentation/>

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Tässä on lippu</name>
    <description>Lippu on asetettu Kartalle</description>
    <Point>
      <coordinates>24.963769,60.165325,100</coordinates>
    </Point>
  </Placemark>
</kml>
```

- <http://kmlvalidator.com/>

Geography Markup Language

- Open Geospatial Consortium kehittämä XML-pohjainen kieli paikkatietojen kuvaukseen
 - <http://www.opengeospatial.org/standards/gml>
- GML on myös ISO standardi ISO 19136:2007
- Versiossa 3.2.1 kaiken kaikkiaan yli 30 XML Schemaa
 - <http://schemas.opengis.net/gml/>
- JHS 162 Paikkatietojen mallintaminen tiedonsiirtoa varten
 - <http://www.jhs-suositukset.fi/suomi/jhs162>
 - Laadittu Paikkatietoasian neuvottelukunnassa (Patine)

GML käsitteitä

- Feature (Kohde) koostuu joukosta propertyjä (ominaisuuksia)
- Ominaisuudet voivat olla joko yksinkertaisia tai monimutkaisia tyyppejä
- Feature association, featureiden välinen suhde
- Käytettävä koordinaattijärjestelmä on valittava
 - Voi käyttää joko standardoituja järjestelmiä nimiavaruuden kautta
 - Koordinaattijärjestelmä voidaan määrittää myös itse

```
<gml:Point gml:id="point96" srsName="urn:x-ogc:def:crs:EPSG::4326"> <gml:pos>-31.936 15.834</gml:pos>  
</gml:Point>
```

- Profiilit rajoittavat käytettävää GML-tekniikkaa. Tarkoituksena mataloittaa tekniikan käyttöönoton kynnystä.
 - Point Profile, GML Simple Features profile

JSON

Yleiskuva

- *JavaScript Object Notation*
- XML:ään verrattavissa oleva kevyt tiedon välityksen tekniikka esimerkiksi selaimen ja palvelimen välille
- Yksinkertainen tekstirakenne
 - Ihmisen luettavissa
 - Helppo parsia ja tuottaa ohjelmallisesti
 - Ohjelmointikielestä riippumaton
- Koska JSON on JavaScriptin tietorakenne, sen käsittely varsinkin selainsovelluksissa on näppärää

JSON vs. XML

- JSON on XML:ää kevyempi tiedonsiirrossa
- JSON-objektit ovat tyypitettyjä toisin kuin XML-data
 - JSON:n tietotyyppejä:
 - String, number, array, boolean
 - XML-data on "vain" merkkijono
- Tiedostomuoto on natiivisti JavaScript-koodia
 - JSON-data ei edellytä erillistä datan parsimista kuten XML-merkkijonot
 - Arvojen lukeminen tapahtuu kuten olion ominaisuuden lukeminen (arvo ~ property)

JSON-olio

- JSON-olio on:
 - Kokoelma nimi/arvo-pareja
 - Lista arvoja
- JSON-olion rakenne:
 - Sisältö rajataan aaltosulkeisiin { }
 - Nimi erotetaan arvosta kaksoispisteellä
 - Nimi/arvo-parit erotetaan toisistaan pilkulla

```
var henkilö = {  
  "etunimi" : "Anna",  
  "sukunimi" : "Malli",  
  "ikä" : 3  
};
```

Datan kuvaus

- Pelkkä data ei sellaisenaan vielä riitä datan hyödyntäjälle
- Mitä formaattia sitten valitaankin käytettävänkin data-formaatti pitää dokumentoida
 - Esimerkiksi XML Schema lyö lukkoon XML-dokumentin tietorakenteen, mutta ei varsinaisesti kuvaa datan merkitystä
- Dataformaatin dokumentointi onnistuu ihan html/pdf/doc/xls-dokumenteilla
- Pitää sisällään kuvauksen datan merkityksestä, mistä data on generoitu
- Käyttöehdot
- Pitää ehkä sisällään valmiita esimerkkejä datan hyödyntämisestä
- Yhteystiedot, keneen voi olla yhteydessä, jos on kysyttävää datasta
- Varsinkin formaateissa joissa ei ole mitään kuvausta datasta, kuten CSV-dataformaatin dokumentointi on hyvin olennaista

REST-palvelut

Tietovaraston julkaisu palveluna

- Eräajotiedostojen sijaan data kannattaisi joissain tilanteissa julkaista palveluna
- Web-palvelimella oleva palvelu pitää sisällään logiikkaa joka palauttaa vain asiakkaan pyytämän datan
- Kaksi perustapaa
 - Representational State Transfer (REST)
 - W3C Web Services (WSDL + SOAP)

Eri ohjelmointimallit (kommunikointi)

REST

- Arkkitehtuurimalli, joka kuvaa kuinka identifioida, esittää ja operoida resursseja
- Korostaa HTTP verbejä, URI:a ja Webin datan esitystapoja
- Toiminta osoitetaan HTTP verbillä
- Ei metadatatukea (ei WSDL:ää, ei XSD:tä)

POX/HTTP

- Yksinkertainen ad-hoc malli välittää XML-muotoista dataa HTTP:n yli
- Yleensä GET/POST, mutta XML-datassa voi olla tarkennus
- Tietyllä tavoin REST & SOAP hybridi

SOAP

- Malli määritellä sovelluksen kommunikointiprotokolla
- Voidaan käyttää mitä tahansa siirtoprotokollaa, esim HTTP, TCP, MSMQ
- vain POST.verbi HTTP:ssä, action määritellään SOAP-viestissä

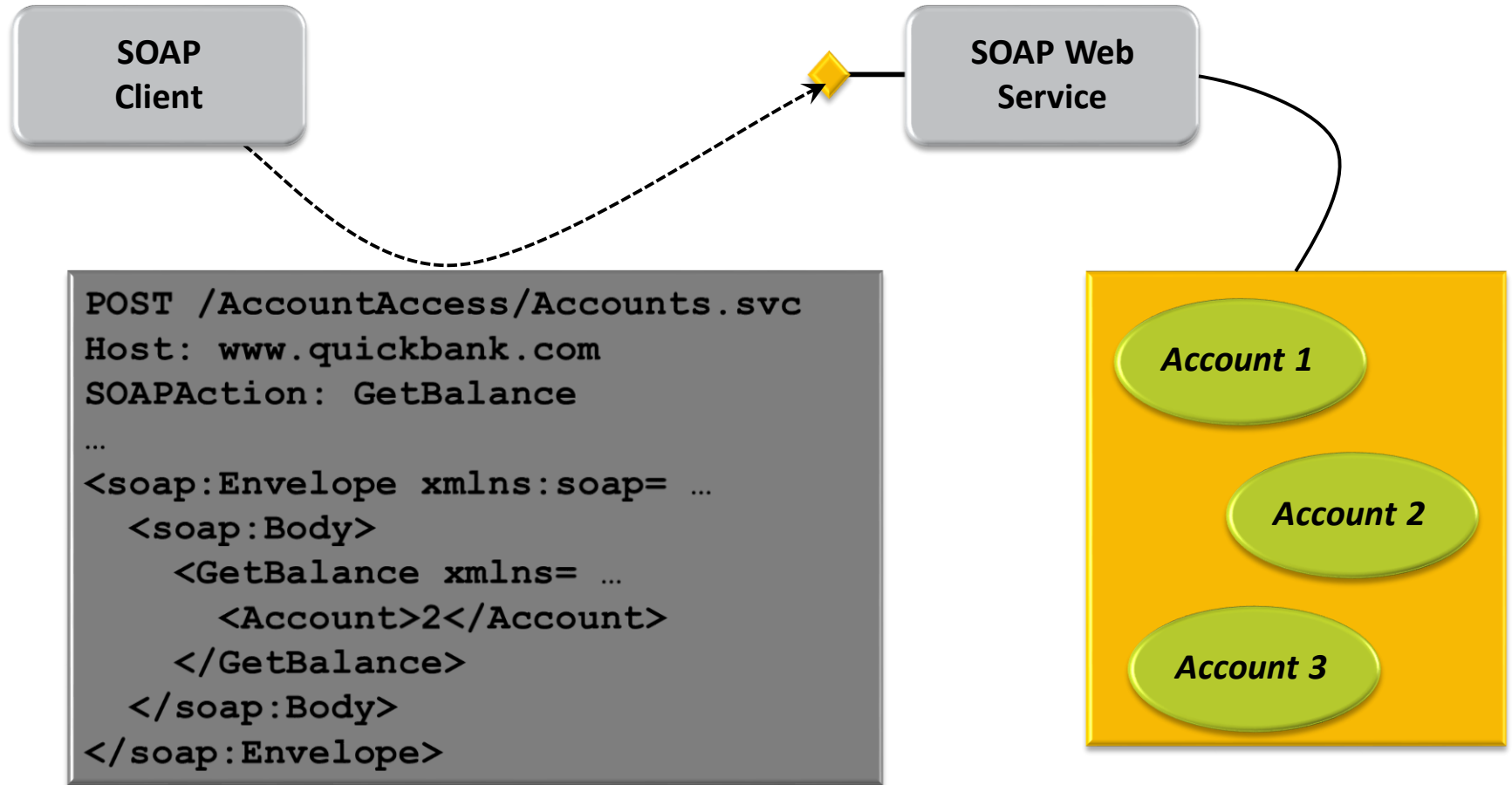
GET

- Web on pääsääntöisesti tehty GET:n varaan
 - GET tarkoittaa "retrieve" (hae esitysmuoto, dokumentti)
 - Ei pitäisi aiheuttaa sivuvaikutuksia (palvelimella)
 - Idempotency
- REST luonnollisesti hyödyntää GET:n voimaa
 - Kun taas SOAP ei käytä sitä lainkaan

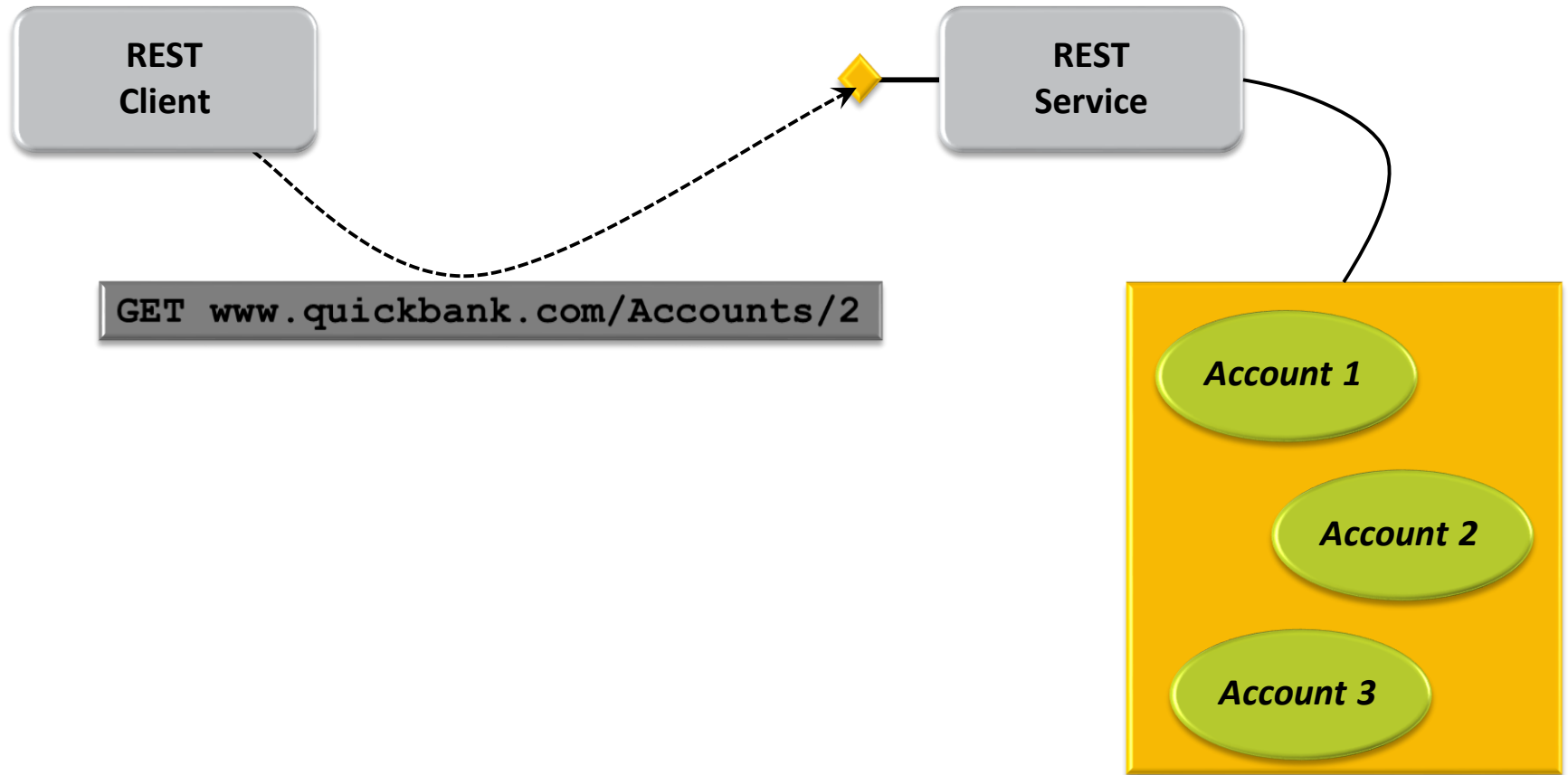
Web Formats

- Laaja käytettävyys edellyttää hyvin ymmärrettyä datan siirtotapaa
 - Jotkut esitystavat ovat yleistyneet (HTML)
 - Muita yleisiä esitystapoja ovat myös RSS ja Atom
- Datan siirtotavaksi on yleistynyt JSON
 - Helppo (de)serialisoida, jolla on merkitystä erityisesti JavaScript-ohjelmoinnissa
 - JSON = JavaScript Object Notation
- Ei standardoitua metadatatukea palvelun rakenteen kuvaamiseen
 - Useita kilpailevia metadata-tekniikkoja joiden yleistyminen kysymysmerkki
 - API-Blueprint, ioDocs, Swagger, WADL (Web Application DL)

SOAP



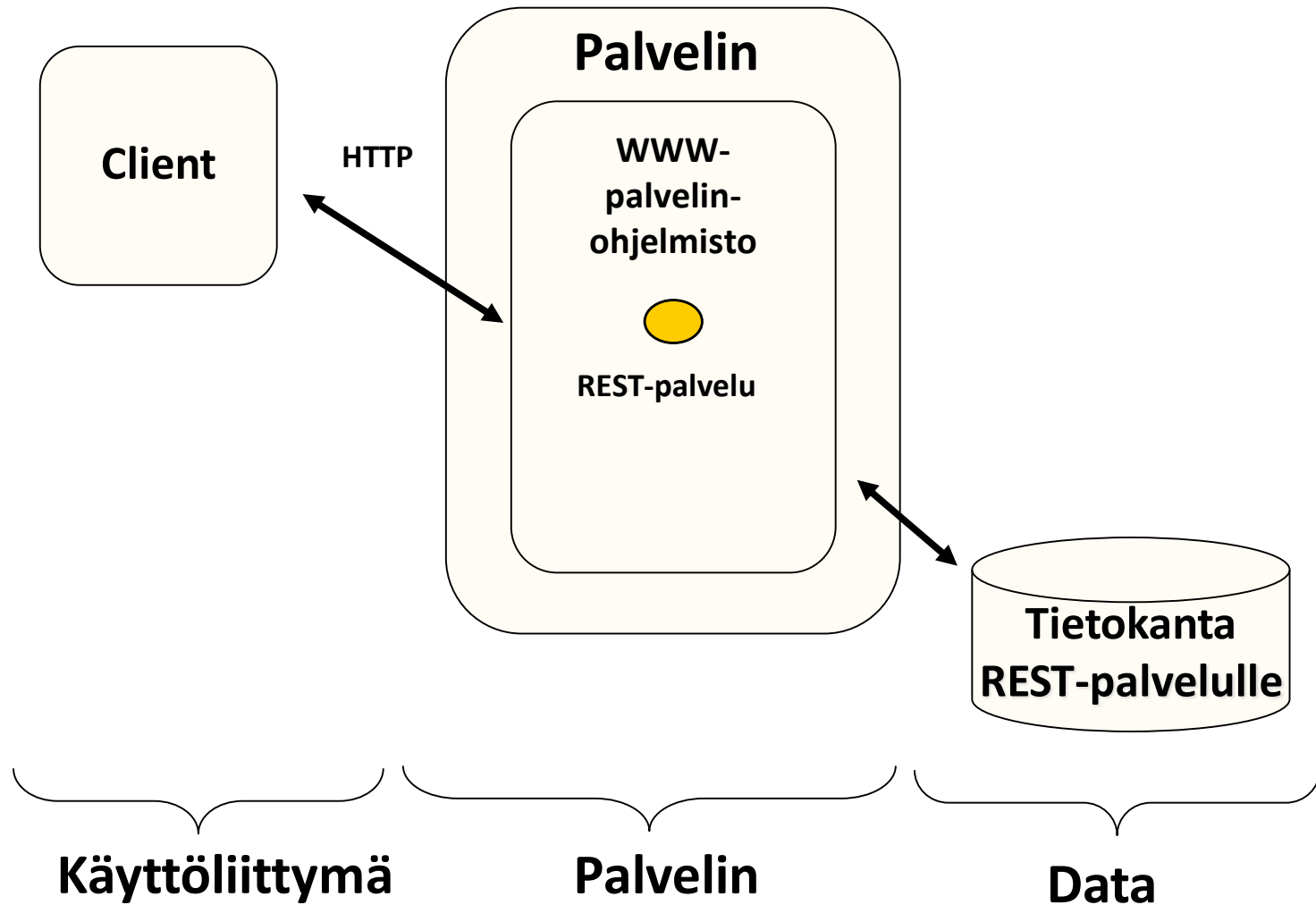
REST



REST arkkitehtuuri

- Kaksi perusperiaatetta
 - Kaikkeen pääsee käsiksi saman rajapinnan kautta
 - Data indentifioidaan URI :lla
- REST :n määrittely
 - *RESTful Web Services*, by Leonard Richardson and Sam Ruby

Rest-palvelun arkkitehtuuri



REST tyylinen kutsu

<http://events.hubi.fi/eventsml/categories>

<http://events.hubi.fi/eventsml/events>

<http://www.hel.fi/palvelukarttaws/rest/v2/unit/?search=lauttasaari+p%E4iv%E4koti>

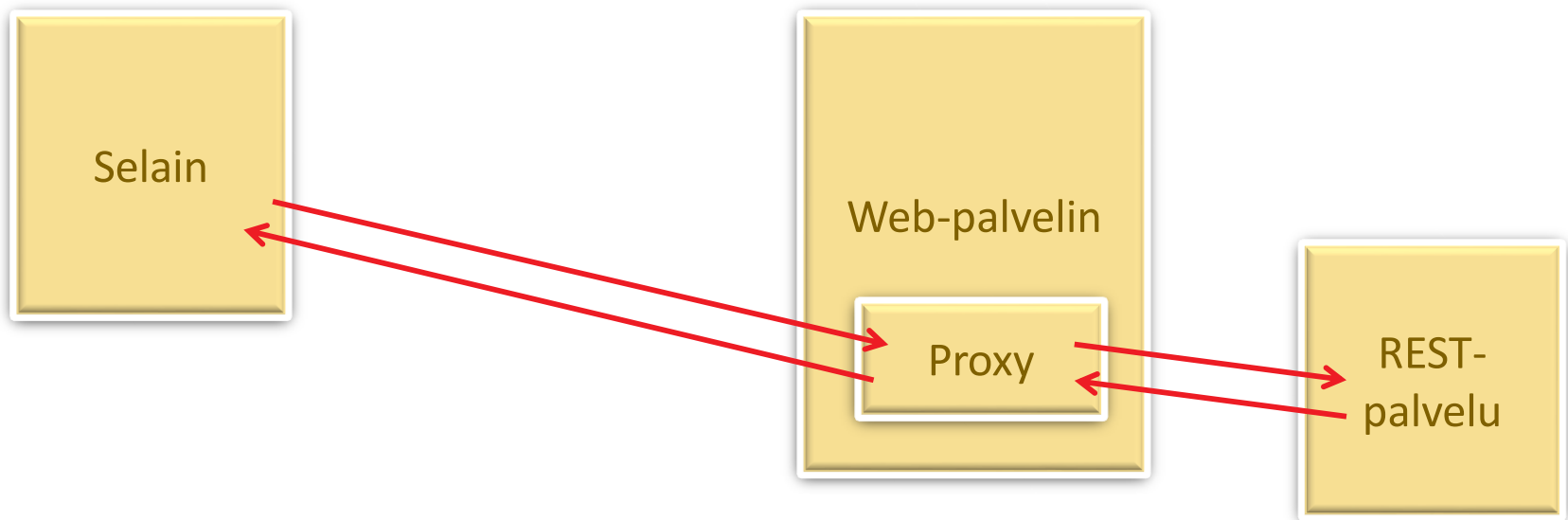
<http://events.hubi.fi/eventsml/events/mevent:17566>

HTTP :n verbit

- REST voi käyttää kaikkia HTTP:n verbejä
 - Erityisesti GET, PUT, POST ja DELETE
- Osan verbeistä (kuten GET) toiminta on tarkoin kuvattu
- Avoimen datan julkaisussa GET olennaisin

Kommunikointi selaimesta

- Selaimissa on tietoturvan takia ns. Same-Origin Policy, joka estää selaimeen ladatusta JavaScriptistä kutsut muille palvelimelle, kuin sille, josta varsinainen JavaScript on ladattu.
- Yksi tapa ratkaista tämä selainsovelluksen tekijän palvelimelle totettuma proxy.



JSONP

- *JavaScript Object Notation with Padding*
- Same-origin policy -tekniikan kiertäminen REST-palveluiden yhteydessä luomalla palautettava JSON-rakenne funktiokutsuna
- Toimintamalli:
 - Sivulle lisätään ohjelmallisesti elementti tyyliä `script src="http://palvelin/jsonp-sisalto.js"`
 - Kyseinen sisältö generoidaan palvelimella ja sen sisältämä JSON-data ympäröidään funktiokutsulla, esimerkiksi `foo({ "nimi": "Anna" });`
 - Lisättyä funktiota kutsutaan sivulla
 - Funktion nimi kuljetetaan usein pyynnön `callback`-nimisenä parametrina, eli URL olisi `http://palvelin/jsonp-sisalto.js?callback=foo`

JSONP

- Avoimen datan julkaisijan toteuttama JSONP:tä tukeva REST-palvelu helpottaa Web-sovelluksien tekijöitä
- Mm. Pääkaupunkiseudun palvelukartalla on JSONP-tuki:
- <http://www.hel.fi/palvelukarttaws/rest/v2/organization/91>
 - `{"id":91,"name_fi":"Helsingin kaupunki","name_sv":"Helsingfors stad","name_en":"City of Helsinki","data_source_url":"www.hel.fi"}`
- <http://www.hel.fi/palvelukarttaws/rest/v2/organization/91?callback=foo>
 - `foo({"id":91,"name_fi":"Helsingin kaupunki","name_sv":"Helsingfors stad","name_en":"City of Helsinki","data_source_url":"www.hel.fi"});`

Open Data Protocol

- Open Data Protocol, ODATA
 - <http://www.odata.org>
- Tuo standardin tavan julkaista tietovarasto REST-palveluna
 - Mahdollistaa CRUD –operaatiot
- OASIS standardoimassa
- Palvelun julkaisuun
 - Kirjastot: .NET (WCF Data Services), Java, node.js, PHP
 - Tietokannat: SQL Server, MySQL , Azure Data
- Asiakaspäähän kirjastoja:
 - HTML5 JavaScript, iOS, Android, WP7, .NET, Java...

Paikkatietopalvelut: GeoRSS

- GeoRSS välittää paikkadataa RSS-syötteessä
- <http://georss.org/>
- Google Map API tukee tätä

```
<feed
....
<author>
<name>SQL Server</name>
</author>
<entry>
<title>Helsingin Pyörä ja Pumppu</title>
<description>Katajanokan laitur, 00160 Helsinki, Tel:09-1234567</description>
<georss:where>
<gml:Point xmlns="http://www.opengis.net/gml"><gml:pos>60.165325 24.963769</gml:pos></gml:Point>
</georss:where>
</entry>
</feed>
```

- <http://feedvalidator.org/>

Paikkatietopalvelut: Web Feature Service - WFS

- Open Geospatial Consortiumin rajapintakuvaus paikkatietojen kyselyyn Internetin yli
- Palvelut voivat palauttaa GML:ää, mutta myös ESRI shapefileja
- Pitää sisällään myös operaatiot kohteiden (feature) ylläpitämiseen
 - Haku, kohteiden luonti, deletointi, päivitys
- Palvelurajanpintana SOAP ja HTTP GET
- Palvelun on vähintään toteutettava seuraavat metodit:
 - GetCapabilities
 - DescribeFeatureType
 - GetFeature, ottaa parametriksi esimerkiksi Bound Box:in
- Useita ohjelmia, jotka tukevat karttatiedon päälle WFS:stä haettujen kohteiden mäppäystä
 - Kosmo GIS, GAIA

Löydettävyyys

- Ei auta, että avoin data web-palvelimen kovalevyn nurkalla kenen tahansa ladattavissa, jos sitä ei kukaan sieltä löydä
- Avoin data pitää olla löydettävissä
- Käytännössä tämä tarkoittaa data-palvelun rekisteröimistä hakemistopalveluihin:
- <http://www.hri.fi/>
- <http://fi.thedatahub.org/>

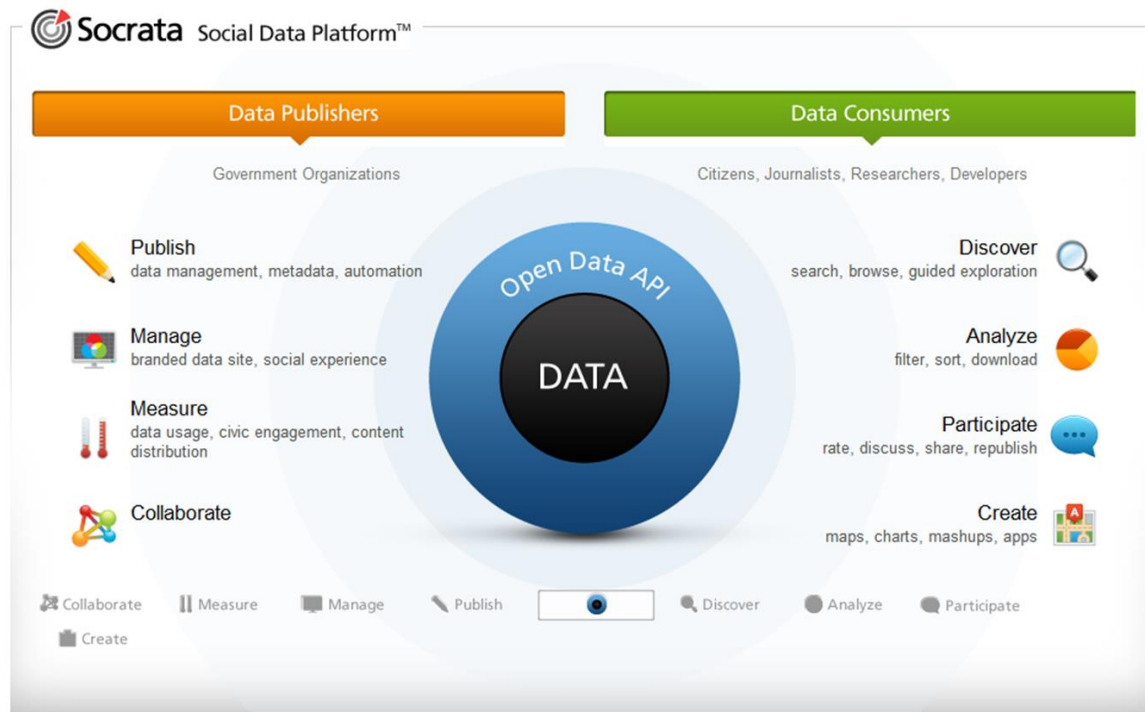
Avoim data ja pilvipalvelut

Pilvipalvelut

- Palveluiden siirtyminen pilveen on ollut selkeä trendi viime vuosina IT-alalla
- Pilvi antaa yhden vaihtoehtoon vikaisietoisuuden ja skaalautumiseen toteuttamiseen kustannustehokkaaseen hintaan
- Avoimen datan julkaisussa pilvessä voisi olla
 - Pelkät eräajotiedostot ladattavissa
 - Itse toteuttama REST-palvelu ja sen taustalla oleva data (PaaS, Platform as a Service –käyttö)
 - Valmiiseen pilvessä olevaan datan julkaisupalveluun ladattu data (SaaS, Software as a Service käyttö)
- Pilvialustoja on mm. Amazon Simple Storage Service (S3), Amazon Elastic Compute Cloud (EC2), Microsoft Windows Azure
- Avoimen datan julkaisu pilvessä voisi antaa julkishallinnolle hyvän mahdollisuuden pilotoida pilviratkaisuja!

Valmiita pilvialustoja avoimen datan julkaisuu

- Buzzdata
 - <http://buzzdata.com/>
 - Datan julkaisuun ja visualisointiin
- Open Government Data Initiative (OGDI)
 - <http://ogdisdk.cloudapp.net/>
 - Microsoftin Windows Azure –pilvialustalla pyörivä avoimen datan julkaisualusta
- Socrata,
 - <http://www.socrata.com>



Avoimen datan visualisointi

Datan visualisointi

- Yksi avoimen datan käyttö on datan visualisoida sitä havainnollisempaan muotoon
- Usein prosessiin liittyy myös tiedonlouhinta (data mining)
 - Tilastollista analyysia, jossa pyritään löytämään olennaisia asioita isosta tietomassasta.
- Datan louhinta ja visualisointi on oma taiteen lajinsa
 - Voi vaatia syvällistä ymmärrystä erilaisista tilastollisista menetelmistä
 - Tilastoanalyysissä ja visualisoinnissa auttavia sovelluksia on lukuisia
 - Joskus toteuttaja voi keksiä aivan uuden tavan visualisoida dataa
- Dataa saatetaan koostaa myös usemmista lähteistä (mashup)
- Big Data –käsitteellä tarkoitetaan isoja ja monimutkaisia tietovarastoja, joiden käsittelyn, analysoinnin ja visualisoinnin mahdollistamiseksi tarvitaan ihan omia ohjelmistoja
- Datan julkaisija voi herättää kiinnostusta julkaisemaan dataansa visualisoimalla dataansa myös itse

Työkaluja

- Tietokantaratkaisut voivat pitää mukanaan valmiita BI-työkaluja (Business Intelligence) datan louhintaan ja visualisointiin
 - Esimerkiksi SQL Server Analysis Services ja Reporting Services, Oracle Business Analytics
- Excel on ollut yksi perinteisimpiä työkaluja datan visualisoinnissa
- Laskentaohjelmistot, kuten Mathematica ja MatLab
- R, GNU General Public lisenssi julkaistu kieli ja ohjelmisto tilastolaskentaan ja graafiseen esitykseen
 - <http://www.r-project.org/>
 - <http://louhos.wordpress.com/linkkeja/>
- Google Chart Tools, datan visualisointiin selaimessa
- Google Fusion Tables
- d3.js open-source JavaScript-kirjasto visualisointiin

Paikkatietodatan visualisointi

- Googlen Maps API tarjoaa hyvät mahdollisuudet
- Paikkatiedot ovat määpättävissä kartan päälle KML-dokumentista tai GeoRSS-feedistä varsin yksinkertaisesti

```
• var myLatLng = new google.maps.LatLng(49.496675, -  
  102.65625);  
  var mapOptions = {  
    zoom: 4,  
    center: myLatLng,  
    mapTypeId: google.maps.MapTypeId.ROADMAP  
  }
```

```
var map = new  
google.maps.Map(document.getElementById("map_canvas"),  
mapOptions);
```

```
var georssLayer = new  
google.maps.KmlLayer(http://www.hel.fi/palvelukartta/kml.as  
px?dohaku=1&haku=vessa');  
georssLayer.setMap(map);
```


Lisätietoja datan visualisoinnista

- <http://www.hri.fi/fi/category/ajankohtaista/visualisointiblogi/>
- <http://www.visualizing.org/>
 - Datavisualisoinnin Web-yhteisö
- <http://flowingdata.com/>
 - Datavisualisointi blog
- <http://datajournalismhandbook.org/1.0/en/>
- <http://www.guardian.co.uk/data>
- <http://talousarvio.juuso.org/>
- <http://bl.ocks.org/mbostock>

Yhteenveto

Palvelun kehittäminen kehittäjäyhteisön kanssa

- Palvelun kehittämisessä kannattaa kuunnella kehittäjiä
 - Ideoita mitä haluttaisiin julkaista
 - Ongelmat
 - Dataformaatti, missä muodossa tieto on
 - Palvelun rajapinta
- Hyvänä esimerkkinä HSL-reittitiedot, joita on kehitetty tiiviisti kehittäjäyhteisön kanssa

Yhteenveto

- Johdanto avoimeen dataan
 - Mitä avoin data on?
 - Miksi avointa dataa pitäisi julkaista?
 - Avoimen datan esimerkkejä
- Dataformaatit
 - Miten valita dataformaatti?
 - Excel, CSV
 - XML ja sen tärkeimmät liitännäistekniikat
 - XML-sanastot
 - JSON
- Miten tuottaa avointa dataa
 - Vaihtoehdot
 - Eräajotiedostot
 - REST-palvelut
 - REST-palvelun toteutus
- Avoimen datan visualisointi

Muita resursseja

- <http://www.julkinendata.fi/>
- <https://www.yhteentoimivuus.fi/>
- <http://www.jhs-suositukset.fi/web/guest>
- <http://www.govdata.eu/>
- <https://www.facebook.com/helsinkiregioninfosshare>
- <https://www.facebook.com/groups/fi.okfn/>