

Joonas Sippola

**Web 2.0**

Erilaisia tekniikoita ja toimintamalleja

Opinnäytetyö

Kesä 2011

Tekniikan yksikkö

Tietojenkäsittelyn koulutusohjelma

Mediaryttäjäyden suuntautumisvaihtoehto



## SEINÄJOEN AMMATTIKORKEAKOULU

### Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Koulutusohjelma: Tietojenkäsittelyn koulutusohjelma

Suuntautumisvaihtoehto: Mediaryrittäjyyden suuntautumisvaihtoehto

Tekijä: Joonas Sippola

Työn nimi: Web 2.0 – Erilaisia tekniikoita ja toimintamalleja

Ohjaaja: Markku Lahti

Vuosi: 2011

Sivumäärä: 48

Liitteiden lukumäärä: 0

---

Tämän opinnäytetyön tavoitteena on selvittää, mitä termi ”Web 2.0” tarkoittaa, minkälaisista elementeistä Web 2.0 koostuu ja millä tekniikoilla Web 2.0 -sivustoja rakennetaan. Työstä myös selviää, kuinka nämä tekniikat ovat vuosien kuluessa kehittyneet sekä kuinka niitä uusilla tavoilla yhdistelemällä on mahdollistettu aivan uudenlaisten, rikkaiden internetsovellusten rakentaminen.

Työssä käydään läpi tekniikoita, kuten PHP ja Ajax, jotka ovat mahdollistaneet internetin nopean kehityksen. Opinnäytetyössä myös sivutaan lähitulevaisuuden mukanaan tuomia mahdollisuuksia.

Lopuksi tavoitteena oli suunnitella ja rakentaa näytesivuja, jotka esittelisivät joitakin Web 2.0:n tärkeimmistä ominaisuuksista.

Avainsanat: Web 2.0, Ajax, rikkaat internetsovellukset

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

## Thesis abstract

Faculty: School of Technology  
Degree programme: Business Information Technology  
Specialisation: Media Entrepreneurship

Author: Joonas Sippola

Title of the thesis: Web 2.0 – Various techniques and approaches

Supervisor: Markku Lahti

Year: 2011

Number of pages: 48

Number of appendices: 0

---

The objective of this thesis was to find out what “Web 2.0” as a term means and what kind of elements Web 2.0 consists of. The reader will learn of the techniques and methods Web 2.0 sites are built with and how this technology has developed over the years, as well as how combining these techniques in new ways has enabled building a completely new kind of application, Rich Web Applications.

The leap forward taken by the internet was made possible by the development of several techniques, such as PHP and Ajax, which will be presented in this thesis. A look at the possibilities brought by the near future will also be taken.

Lastly, the purpose was to design and build sample pages, which would showcase some of the most prominent features of Web 2.0.

Keywords: Web 2.0, Ajax, Rich Internet Applications

## SISÄLTÖ

Opinnäytetyön tiivistelmä.....	2
Thesis abstract.....	3
SISÄLTÖ.....	4
Käytetyt termit ja lyhenteet .....	6
Kuvioluettelo.....	7
<b>1 JOHDANTO .....</b>	<b>9</b>
1.1 Työn taustaa .....	9
1.2 Työn tavoite .....	9
1.3 Työn rakenne .....	10
<b>2 INTERNETIN KEHITYSASKELEET .....</b>	<b>11</b>
2.1 Kuvauskielet.....	13
2.1.1 HTML .....	13
2.1.2 XML .....	13
2.2 Skriptikielet.....	14
2.2.1 JavaScript .....	14
2.2.2 PHP .....	17
2.3 Liitännäiset.....	19
2.3.1 Flash .....	19
2.3.2 Silverlight .....	20
2.4 Yhdistelmätekniikat .....	21
2.4.1 DHTML.....	21
2.4.2 Ajax.....	22
<b>3 MITÄ WEB 2.0 TARKOITTAA? .....</b>	<b>23</b>
3.1 Rikkaat internetsovellukset .....	24
3.2 Palvelukeskeinen arkkitehtuuri.....	25
3.2.1 Pilvipalvelut .....	26
3.2.2 Yhdistelmäpalvelut.....	30
3.3 Sosiaalinen verkko .....	32
3.3.1 Blogit.....	33

3.3.2 Sosiaaliset verkostot .....	34
3.4 Web 2.0 -käsitteen tarpeellisuudesta herännyt kritiikki.....	35
4 TULEVAISUUS .....	36
4.1 HTML5 .....	36
4.2 Web 2.0:n jälkeen .....	37
5 NÄYTESIVUJEN RAKENTAMINEN.....	38
5.1 Suunnittelu .....	38
5.2 Toteutus .....	38
5.3 Tulokset .....	40
6 YHTEENVETO.....	43
LÄHTEET .....	44

## Käytetyt termit ja lyhenteet

<b>CSS</b>	Cascading Style Sheets, tyyliohjeet, kokoelma sääntöjä, joilla kuvataan web-sivun ulkoasua (Korpela & Linjama 2005, 39).
<b>DOM</b>	Document Object Model, ohjelmointirajapinta, jonka avulla JavaScriptillä voidaan käsitellä web-sivun sisältöä puumaisena rakenteena (Ek & Norén 1999, 9).
<b>RSS</b>	Really Simple Syndication, syöte, jonka avulla useasti päivittyvän web-sivun, kuten esimerkiksi blogin, seuraaminen helpottuu (Kilpi 2006, 19).
<b>SQL</b>	Structured Query Language, standardoitu kyselykieli, jolla käsitellään relaatiomallisia tietokantoja (Rantala 2005, 252).
<b>W3C</b>	World Wide Web Consortium, kansainvälinen yhteenliittymä, joka ylläpitää ja kehittää web-standardeja ja ohjeita, joiden tarkoituksena on varmistaa webin pitkäjänteinen kasvu (W3C 2009).
<b>Web</b>	World Wide Webistä käytetään tässä opinnäytetyössä keskenään vaihdettavasti sanoja web ja internet.

## Kuvioluettelo

KUVIO 1. Staattinen web-sivu.....	12
KUVIO 2. Dynaaminen web-sivu.....	12
KUVIO 3. jQuery:n käyttöaste prosenteissa suosituimpien sivustojen kohdalla.....	15
KUVIO 4. JavaScript-kirjastojen käyttöaste valvotuilla verkkosivuilla.....	16
KUVIO 5. Kokoelma Web 2.0 -sivustojen logoja .....	23
KUVIO 6. Joli OS:n työpöytänäkymä .....	25
KUVIO 7. Pilvipalvelut liiketoimintaprosessien tukena .....	27
KUVIO 8. Spotifyn asiakasohjelmisto.....	29
KUVIO 9. Tilannehuone.fi:n hälytyskartta .....	31
KUVIO 10. Flightradar24.com:n reittilentokartta.....	32
KUVIO 11. Sosiaalisia Web 2.0 -sivustoja ryhmiteltynä aiheen mukaan.....	33
KUVIO 12. HTML5:n logo .....	36
KUVIO 13. Kaavio kehityksestä kohti Web 4.0:aa .....	37
KUVIO 14. Luotu Facebook-sivu.....	40
KUVIO 15. Blogin etusivu .....	41

KUVIO 16. Blogin kommenttisivu .....	41
KUVIO 17. Arvosanasovelluksen alku tila .....	42
KUVIO 18. Arvosanasovellus esittää arvosanojen keskiarvon .....	42



# 1 JOHDANTO

## 1.1 Työn taustaa

Web 2.0 on ilmiö, josta ei ole voinut välttää kuulemasta. Sen osakseen saama kiinnostus on kasvanut viimeisten vuosien aikana eksponentiaalisesti. Termi ”Web 2.0” esiteltiin ensimmäisen kerran vuonna 2004, mutta silloin tuskin kukaan osasi vielä arvata, kuinka suuri ilmiöstä vielä tulisi. Julkisuuteen termin toivat Tim O’Reilly ja Dale Dougherty (Tirronen 2008, 13).

Internet on murrostilassa. Uudetkin verkkosivut tuntuvat vanhentuneilta, jos ne on toteutettu ns. vanhalla tekniikalla.

## 1.2 Työn tavoite

Tämän opinnäytetyön tavoitteena on antaa kuva internetin nykytilasta, lähitulevaisuudesta ja nämä mahdollistaneista tekniikoista. Tekniikat esitellään aluksi yksitellen ja sen jälkeen yhdistelminä. Tavoitteena on myös avata käsite Web 2.0 pienempiin osa-alueisiin ja käydä näitä alueita läpi esimerkkien avulla.

Teoriaosuuden lisäksi tavoitteena on rakentaa joitakin esiteltyjä tekniikoita soveltamalla Web 2.0 -näytesivuja. Näiden sivujen tarkoitus on havainnollistaa, mitä Web 2.0 käytännössä tarkoittaa.

### **1.3 Työn rakenne**

Luvussa 2 esitellään uuden sukupolven verkkosivut mahdollistaneita tekniikoita ja niiden kehitystä. Luvussa 3 kerrotaan, mitä käsitteellä Web 2.0 tarkoitetaan ja minkälaisista elementeistä se koostuu. Luvussa 4 esitellään lähitulevaisuuden näkymiä. Luvussa 5 rakennetaan Web 2.0:n tärkeimpiä ominaisuuksia esitteleviä näytesivuja sekä kerrotaan, kuinka tämä onnistui. Yhteenveto on sijoitettu lukuun 6.

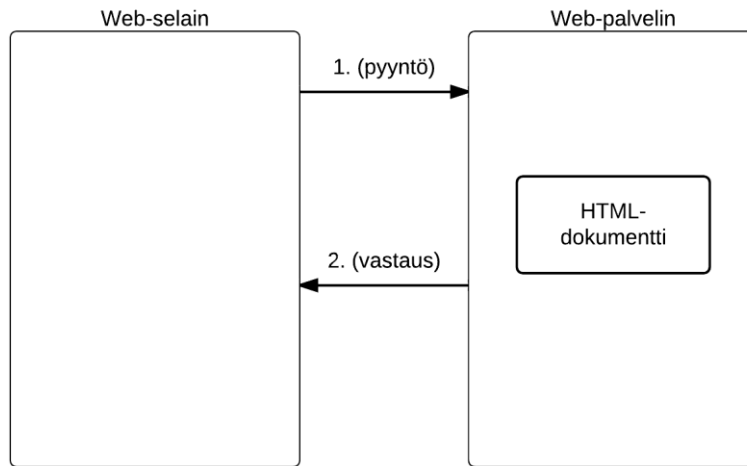
## 2 INTERNETIN KEHITYSASKELEET

Aluksi kaikki internetsivut olivat staattisia. Käyttäjän web-selain lähetti sivupyynnön palvelimelle, joka vastasi siihen palauttamalla pyydetyn sivun (Kuvio 1). Sivut eivät sisältäneet mitään ylimääräistä. Ne olivat käytännössä vain tekstidokumenttien sähköisiä kopioita. (Asleson & Schutta 2007, 3.)

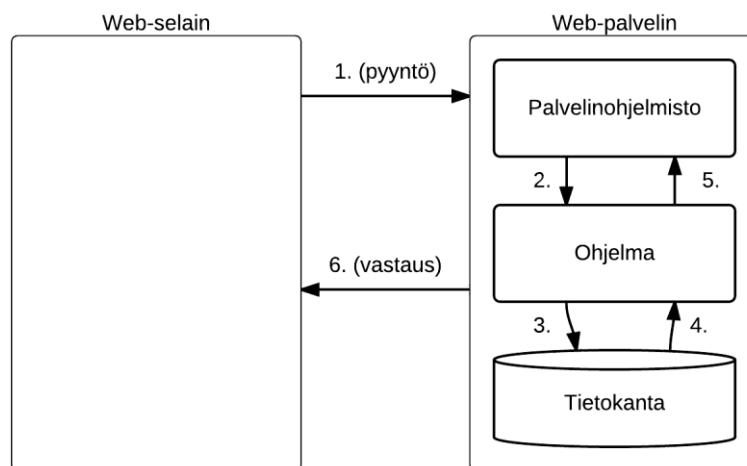
**Staattinen web-sivu.** Staattinen sivu säilyy muuttumattomana, kunnes sen ylläpitäjä päivittää sinne uutta sisältöä. Staattinen sivu saattaa kuitenkin sisältää dynaamisia elementtejä, jotka muuttuvat käyttäjän toimien seurauksena, mutta muutokset tapahtuvat aina silti vain käyttäjän omalla päätelaitteella. Evästeiden tyhjentäminen ja sivun uudelleenlataaminen nollaa muutokset. (Rantala 2005, 3.)

**Dynaaminen web-sivu.** Dynaamisen sivun sisältö taas voi muuttua käyttäjän tai ylläpidon toimista riippumatta ja säilyttää muutoksensa esimerkiksi tietokannan avulla. Sivun dynamiikka voi myös toteutua vain palvelintasolla, jolloin sivu näytetään staattisen sivun tavoin käyttäjän ladattua sen kerran selaimeensa, mutta uudelleenlataamisen jälkeen sivu saattaakin olla erinäköinen. (Rantala 2005, 4.)

Dynaamisen web-sivun muodostaminen palvelimella alkaa web-selaimen pyynnöstä (Kuvio 2). Web-palvelin huomaa esimerkiksi pyynnössä (1) esiintyvistä ".php"-tarkenteesta, että sen tulee käynnistää tietty ohjelma, tässä tapauksessa PHP-tulkki (2). Ohjelma voi mahdollisesti pyytää tietoja tietokannasta (3). Kun tiedot on saatu (4), ohjelma tuottaa (5) HTML-kieleen pohjautuvan web-sivun, jonka web-palvelin välittää vastauksenaan (6) web-selaimelle. (Rantala 2005, 4.)



KUVIO 1. Staattinen web-sivu. (Rantala 2005, 3.)



KUVIO 2. Dynaaminen web-sivu. (Rantala 2005, 4.)

**Web-sovellus.** Tietynlaisten tekniikoiden avulla on myös mahdollista luoda web-sivuja osittain tai kokonaan dynaamisesti. Jos tällainen pääsääntöisesti dynaamisesti luotu web-sivusto on rakennettu jotain tiettyä käyttötarkoitusta varten, voidaan jo puhua web-sovelluksesta. (Rantala 2005, 4, 74.)

## 2.1 Kuvauskielet

Kuvauskielet, tai merkintäkielet, ovat käsitteellisiä menetelmiä, joilla tekstisisällölle voidaan antaa rakenne (Korpela & Linjama 2005, 71).

### 2.1.1 HTML

HTML on standardoitu kuvauskieli. Se on alusta lähtien ollut web-sivujen rakentamisen keskeisin tekniikka. Lyhenne tulee sanoista *Hypertext Markup Language*, eli hypertekstin merkintäkieli. HTML:n avulla kuvataan verkkosivujen rakennetta ja muotoa, sekä haluttaessa myös ulkoasua. Ulkoasun kuvaaminen ei kuitenkaan ole suositeltavaa. Tätä tarkoitusta varten on kehitetty CSS-tyyliohjeet. Pelkällä HTML:llä tuotetut sivut ovat staattisia. (Korpela & Linjama 2005, 70; Salo 2010, 58.)

HTML-dokumentti sisältää tekstisisällön lisäksi sen loogisen rakenteen osoittavat merkinnät, sekä viittauksia muuntyyppisiin tiedostoihin, kuten esimerkiksi kuviin. HTML:n tärkein piirre oli alusta alkaen hyperteksti, eli linkit muihin dokumentteihin. (Korpela & Linjama 2005, 70; Salo 2010, 58.)

### 2.1.2 XML

XML tulee sanoista *eXtensible Markup Language*, eli laajennettava merkintäkieli. HTML:stä se eroaa siinä, että XML ei ole valmis malli tai dataformaatti, vaan kieli, joka mahdollistaa kehittäjän itse määrittelemien kuvauselementtien käytön (Ek & Norén 1999, 3–4). Määriteltyjen elementtien avulla niiden sisältämän tiedon merkitys voidaan sisällyttää web-dokumentteihin (Sharma 2011). Esimerkiksi kemiallisia tai matemaattisia kaavoja ei pelkän HTML:n keinoin voida esittää. XML:n avulla käyttäjän taas on mahdollista määrittää omia elementtejä, joilla kuvata tätä informaatiota (Ek & Norén 1999, 129).

## 2.2 Skriptikielet

Skriptikieliä käytetään erityisesti sovellusten toimintojen laajentamiseen. Skriptin pituus voi vaihdella yhdestä rivistä aina kokonaiseen laajaan sovellukseen saakka. Skriptikielellä kirjoitettua ohjelmaa ei tarvitse perinteisempien ohjelmointikielten tapaan kääntää suoritettavaan muotoon, vaan skriptin voi suorittaa sellaisenaan. Tarvitaan vain sovellus, joka sisältää skriptikielen tulkin. (Peltomäki 2001, 9.)

### 2.2.1 JavaScript

JavaScriptistä on tullut tärkeä osa viime vuosina kehitettyjä verkkopalveluita. Se on skriptikieli, joka suunniteltiin helpottamaan internetsovellusten kehittämistä. Kaikki nykyaikaiset web-selaimet pystyvät suorittamaan JavaScript-skriptejä ilman asennettavia lisäosia. JavaScriptin yleisin käyttötarkoitus on toiminnallisuuksien ja vuorovaikutusmahdollisuuksien lisääminen verkkosivuille ja -palveluihin. JavaScript mahdollistaa myös web-palvelun resurssi-intensiivisten osien suorittamisen paikallisesti. Tämä säästää arvokasta palvelinkapasiteettia. (Asleson & Schutta 2007, 6; Salo 2010, 58–59.)

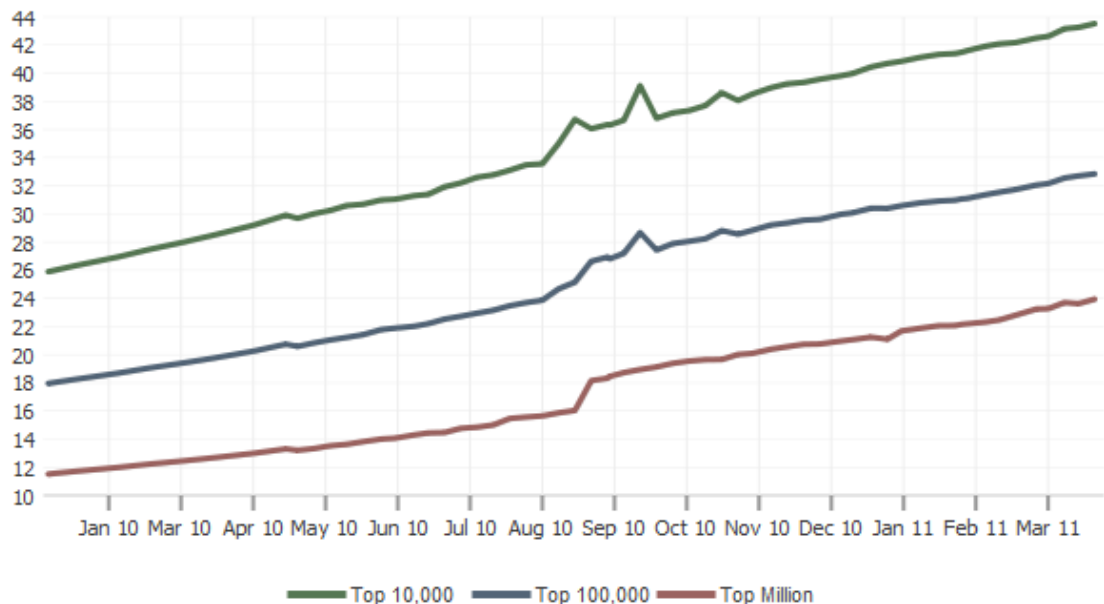
JavaScript on saanut osakseen kritiikkiä sen kehitystyökaluista, epähavainnollisista virheilmoituksista ja puuttuvista virheidenjäljittimistä (Asleson & Schutta 2007, 6). Näistä huolimatta se on silti tehokas työkalu luotaessa dynaamista toiminnallisuutta muuten staattisille sivuille. JavaScriptiä voi käyttää osana Ajaxia myös aidosti dynaamisten sivujen luomiseen. JavaScript tunnetaan standardoituna nimellä ECMAScript. Ongelmat tavaramerkkien kanssa estivät JavaScript-nimen käytön standardissa. (Krill 2008; Ecma International 2009, 7.)

**JavaScriptin historia.** Netscape julkaisi kehittämänsä JavaScriptin vuonna 1995. Kehityksen aikana sitä kutsuttiin ensin Mochaksi ja myöhemmin LiveWireksi ja LiveScriptiksi ennen lopullista nimeämistä JavaScriptiksi. (Asleson & Schutta 2007, 6.)

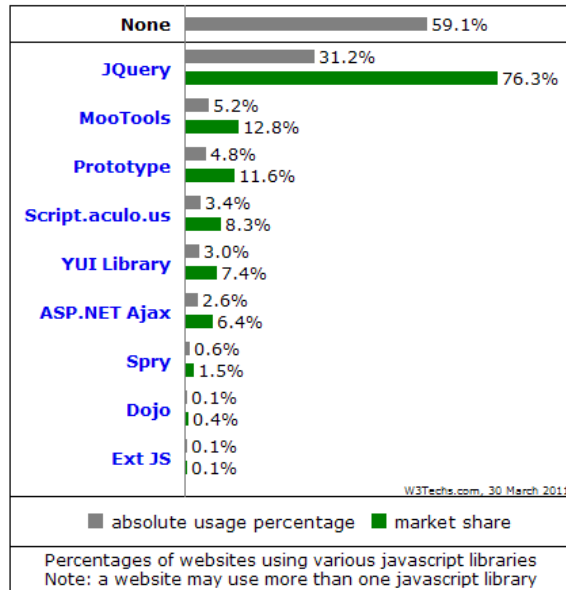
**JavaScript-kirjastot.** Yksi JavaScriptin perusajatuksista on käytön helppous. JavaScriptin opiskelun voi aloittaa lainaamalla aluksi yksinkertaisen, muutaman rivin pituisen skriptin joltain internetin lukuisista JavaScript-ohjesivuista. Kun sen toimintaperiaatteen ymmärtää, sitä voi ryhtyä muokkaamaan paremmin omiin tarpeisiin sopivaksi. (Krill 2008.)

Internetistä löytyy myös yleiseen käyttöön tarkoitettuja JavaScript-kirjastoja. Näiden tarkoituksena on helpottaa web-kehittäjän työtä tarjoamalla yleisimpiä sovelluksia vapaaseen käyttöön. Kirjastoihin on pyritty kehittämään usein vastaantuleviin tarpeisiin ja ongelmiin laadukkaat ja tehokkaat ratkaisut. (Salo 2010, 59.)

jQuery on yksi suosituimmista JavaScript-kirjastoista. Yli 40 prosenttia 10 000 suosituimmasta internetsivusta käyttää sitä (Kuvio 3) (BuiltWith Technology Usage Statistics 2011). Näihin kuuluvat muun muassa Nokia Ovi, Coca-Cola, Nike ja Adidas (Salo 2010, 59). Muita suosittuja kirjastoja ovat muun muassa MooTools, Prototype, Script.aculo.us ja Yahoo User Interface Library (Kuvio 4) (W3Techs 2011).



KUVIO 3. jQueryn käyttöaste prosenteissa suosituimpien sivustojen kohdalla. (BuiltWith Technology Usage Statistics 2011.)



KUVIO 4. JavaScript-kirjastojen käyttöaste valvotuilla verkkosivuilla. (W3Techs 2011.)

**JavaScript ja Microsoft.** Microsoftin oma vastine JavaScriptille on VBScript-skriptikieli. Se kuitenkin toimii ilman liitännäisiä vain Internet Explorer -selaimella, joten sen käyttämistä tulee harkita (Sharma 2011). Microsoftin JavaScript-tuki poikkeaa normaalista. Se ei lisenssisyistä itse asiassa edes varsinaisesti tue JavaScriptiä, vaan Jscriptiä, joka nimeään lukuun ottamatta on käytännössä identtinen JavaScriptin kanssa. (Krill 2008.)

**JavaScript ja Java.** JavaScriptiä ei tule sekoittaa Javaan. Samantyylisten nimien vuoksi on todella yleistä olettaa JavaScriptin olevan yksinkertaistettu versio Javasta. Näillä kahdella ei kuitenkaan ole mitään tekemistä keskenään, vaan LiveScript-nimen vaihtamisessa JavaScriptiksi kyseessä oli pikemminkin markkinointikikka, joilla Netscape sai uudelle skriptikielelleen lisää julkisuutta Sun Microsystemsin juuri julkaiseman, jo suosiota saaneen Javan nimellä. (Flanagan 2006, 2.)



### 2.2.2 PHP

PHP on palvelimella käytettävä ohjelmointikieli, jonka avulla kehittäjät voivat luoda dynaamisia web-sivuja. Se on tehokas ja joustava kieli, jota käytettäessä useiden projektien tuotantoaika on huomattavasti lyhyempi kuin muita menetelmiä käytettäessä. Nimi PHP tulee sanoista *PHP: Hypertext Preprocessor*. (Zandstra 2001, 22.)

PHP-koodi osaa tehdä kyselyjä monenlaisista tietokannoista, luoda kuvia, lukea ja kirjoittaa tiedostoja sekä keskustella etäpalvelimien kanssa. Se on Zandstran (2001, 20–21) mukaan yksi maailman yleisimmin käytetyn palvelinohjelmiston, Apache HTTP Serverin, suosituimpia moduuleita. (Rantala 2005, 9–10.)

PHP:n lähdekoodi on vapaasti saatavilla. Tämän ja sen helpon opittavuuden, monipuolisuuden ja tehokkuuden ansiosta PHP:n suosio on kasvanut nopeasti. PHP toimii useissa eri käyttöjärjestelmissä ja useiden palvelinohjelmistojen sekä tietokantojen kanssa. (Zandstra 2001, 22; Rantala 2005, 9–10.)

**PHP:n historia.** PHP:n kehitystyö alkoi vuonna 1994. PHP koostui vielä silloin joukosta web-julkaisumakroja ja lyhenne PHP johdettiin nimestä *Personal Home Page Tools*. Myöhemmin PHP:hen lisättiin Forms Interpreter -komponentti, ja koko työkalupaketti nimettiin uudelleen PHP/FI:ksi. (Zandstra 2001, 20.)

Vuonna 1997 PHP alkoi herättää kiinnostusta suuremmassa kehittäjäyhteisössä. Joukko ohjelmoijia ryhtyi projektiin, jonka tarkoituksena oli kirjoittaa PHP uusiksi. Tämä, jo kolmas uudelleenkirjoitus, teki PHP:stä vihdoinkin vakavasti otettavan ohjelmointikielen. Kolmannen version julkistamisen jälkeen sen suosio kasvoi huomattavasti. PHP4 toi mukanaan vielä joukon uusia piirteitä, jotka helpottivat ohjelmointia sekä nopeuttivat koodin suoritusta suuresti. (Zandstra 2001, 20.)

**PHP-sovelluskehikset.** PHP-sovelluskehikset (*PHP Frameworks*) tarjoavat pohjan, jonka avulla web-sovellusten kehittäminen on helpompaa kuin ”puhtaalta pöydältä” aloitettaessa. Ne mahdollistavat nopean sovelluskehityksen (*Rapid Application Development*), joka säästää aikaa ja vähentää itseään toistavan koodin kirjoit-

tamista. Sovelluskehukset myös auttavat rakentamaan vakaampia sovelluksia ja avustavat oikeaoppisten tietokantayhteyksien luomista, mikä helpottaa suuresti varsinkin aloittelevia web-kehittäjiä. (Agarwal 2010; Reyes 2011.)

PHP on joustava kieli, joka antaa sitä käyttävälle kehittäjälle anteeksi monenlaisia pieniä virheitä. Sovellus saattaa päällisin puolin toimia edelleen hyvin, mutta kehittäjän tietämättä sovellukseen on saattanut ilmaantua suuriakin tietoturva-aukkoja. PHP-sovelluskehukset auttavat minimoimaan näidenkin uhkien mahdollisuuksia. (Reyes 2011.)

Muun muassa seuraavat sovelluskehukset ovat varsin suosittuja:

- **Zend Framework** keskittyy Web 2.0 -tyylisten sovellusten rakentamiseen, mutta vaatii käyttäjältään laajan PHP-osaamisen. Sen tarjoamat laajat tukimahdollisuudet ovat tehneet siitä yhden suosituimmista sovelluskehyksistä.
- **CakePHP** on uudempi kehys, joka kasvattaa suosiotaan nopeasti. Se on keskittynyt nopeaan sovelluskehitykseen ja on yksinkertainen käyttää. Yksinkertaisuudestaan huolimatta se pystyy skaalautumaan isommankin sovelluksen tarpeisiin.
- **CodeIgniter** on nopea ja helppokäyttöinen sovelluskehys. Se on aloittelijaystävällinen ja tarjoaa laajat tukimahdollisuudet, myös video-oppaiden muodossa.
- **Yii** on komponenttipohjainen sovelluskehys, joka on suunnattu erityisesti suuremman skaalan Web 2.0 -sovellusten rakentamiseen.
- **Symfony** on suunnattu pikemminkin yritys- kuin yksityiskäyttöön. Siinä on monia ominaisuuksia, mutta se ei välttämättä pärjää nopeudessa muille kehyksille.
- **Fat Free, Konstrukt, LightVC, Adroit, Tekuna** ja **UltraLite** ovat sovelluskehysiksiä, joiden suunnittelussa on panostettu ennen kaikkea keveyteen ja minimaalisiin järjestelmävaatimuksiin.

(PHPFrameworks.com 2007; Agarwal 2010; Reyes 2011.)

## 2.3 Liitännäiset

Liitännäinen (*plugin*) on web-selaimen kanssa vuorovaikutuksessa toimiva sovel-  
lus, joka mahdollistaa toimintoja, joihin selain ei itsessään pysty. Liitännäisten  
asentaminen vaatii käyttäjän omaa aktiivisuutta. Liitännäiset kasvattavat myös tie-  
toturvariskiä, vaativat ajoittain päivityksiä ja huonosti toimiessaan haittaavat selai-  
men toimintaa. (Salo 2010, 57.)

Liitännäisten tarpeellisuus tulevaisuudessa vähenee uuden HTML5-standardin  
myötä (Salo 2010, 58). Toisaalta Google on paketoanut Flash-liitännäisen Chrome-  
selaimensa mukana tulevaksi vakio-osaksi (Räihä 2010). Liitännäinen ei ole sama  
asia kuin selaimen lisäosa (*add-on*) tai laajennus (*extension*), jotka asennetaan  
osaksi selainta.

### 2.3.1 Flash

Flash on suosittu multimedia-alusta. Flash-liitännäinen mahdollistaa Flash-  
kehitysympäristössä tuotettujen esitysten ja sovellusten toistamisen web-  
selaimella. Flashin avulla luotuja dynaamisia web-sovelluksia voi toiminnallisuu-  
tensa puolesta hyvin verrata tavanomaisiin sovelluksiin (Asleson & Schutta 2007,  
9). Googlen omistama YouTube käyttää Flashia, kuten myös monet muut videosi-  
vustot sekä selainpohjaiset verkkopelit ja virtuaaliyhteisöt. (Paananen 2008, 7;  
Salo 2010, 57.)

Flash on oman valmistajansa tekemän tutkimuksen mukaan maailman suosituin  
selainliitännäinen ollen asennettuna 99 prosenttiin käyttäjien koneista Yhdysval-  
loissa, Kanadassa, Saksassa, Ranskassa, Isossa-Britanniassa, Australiassa ja  
Uudessa-Seelannissa. Tätä tutkimustulosta ei kuitenkaan kannattane hyväksyä  
absoluuttisena totuutena. (Salo 2010, 57.)

**Flashin historiaa.** FutureWave-niminen yhtiö julkaisi kesällä 1996 Flashin esias-teen, FutureSplash Animator -nimisen tuotteen. Tuote myytiin yrityskaupan myötä Macromedialle, joka nimesi tuotteen uudelleen ensin Shockwave Flashiksi ja ly- hensi sen myöhemmin pelkäsi Flashiksi. Nykyään Flashin omistaa Adobe. (Asle- son & Schutta 2007, 9; Paananen 2008, 7.)

**Apple ja Flash.** Apple on suhtautunut Flashiin viime aikoina nuivasti. Se on estä- nyt Flashin julkaisemisen iOS-käyttöjärjestelmää käyttäville laitteilleen, eli iPod Touchille, iPhoneille ja iPadille. Lisäksi vuoden 2010 loppupuolella Apple ilmoitti, ettei se tule enää toimittamaan Flash-liitännäistä valmiiksi asennettuna käyttöjär- jestelmä Mac OS X:n mukana. Apple perustelee muutosta tietoturvalla. Sen mu- kaan paras tapa tukea Flashia on varmistaa, että käyttäjät lataavat ja asentavat varmasti tuoreimman version suoraan Adobelta (Masalin 2010). Symantec- tietoturvaohjelmistovalmistajan julkaiseman tiedon mukaan Flashin tietoturvatoteu- tus oli vuoden 2009 huonoimpia (Jobs 2010).

Flash on suljettu formaatti. Sen kehitys ja levitys riippuu Applen toimitusjohtaja Steve Jobsin (2010) mukaan täysin Adobesta. Flashia tarvittiin aikoinaan HTML- sivuihin upotettujen videoiden toistamiseen, mutta tämä on mahdollista myös HTML5:n avulla. YouTube sisältää arviolta 40 prosenttia webin videomateriaalista, ja sitä voi käyttää nykyisin ilman Flash-liitännäisen asentamista. (Jobs 2010.)

### 2.3.2 Silverlight

Microsoft on kehittänyt Flashin kanssa kilpailevan Silverlightin. Silverlightin kehi- tystyössä käytetään Microsoftin .NET-tekniikoita, joiden käyttäjäkunta yritysmaail- massa on laaja. Silverlight toimii Windows- ja Mac OS X -käyttöjärjestelmissä, mutta sille on kuitenkin olemassa myös avoimen lähdekoodin vaihtoehto, Moon- light. Microsoftin ja Novellin yhteistyössä toteuttaman Moonlightin avulla Silver- light-tekniikalla toteutetut toiminnallisuudet saadaan myös muun muassa Linux- käyttöjärjestelmiin. (Salo 2010, 57.)

Silverlight ei kuitenkaan ole saanut yhtä suurta huomiota eikä käyttäjämäärää kuin Flash sen kehittyneemmästä teknisestä toteutuksesta huolimatta. Tulevaisuudessa Microsoft painottaakin tekniikan kehittämistä mobiilialusta Windows Phonelle, korostaen HTML:n asemaa ainoana todellisena alustariippumattomana ratkaisuna. (Foley 2010)

## **2.4 Yhdistelmätekniikat**

Yhdistelmätekniikat eivät ole yksittäisiä standardoituja tekniikoita, vaan pikemminkin itse koottuja, epävirallisia kokoelmia tekniikoista, jotka toimivat hyvin yhdessä. Niiden vahvuus on niiden rakentuminen vanhempien, jo olemassa olevien tekniikoiden varaan. Näin kehittäjien ei tarvitse opetella uusia ohjelmointikieliä ja aikaa säästyy. (Asleson & Schutta 2007, 11–14.)

### **2.4.1 DHTML**

Vuonna 1997 web-kehittäjät saivat käyttöönsä uuden menetelmän: dynaamisen HTML:n eli DHTML:n. Tekniikka koostuu JavaScriptista sekä CSS-tyyliohjeiden ja DOM-mallin käsittelystä. Yhdistelmän avulla kehittäjät voivat muuttaa web-sivun rakennetta ja sisältöä sivun ollessa jo ladattuna käyttäjän selaimessa, päivittämättä sitä erikseen palvelimelta. Nimestään huolimatta DHTML:n avulla luodut sivut eivät kuitenkaan ole aidosti dynaamisia, vaan dynaamisuus toteutuu staattisella sivupohjalla. DHTML:n voidaan ajatella olevan Ajaxin esiaste. (Asleson & Schutta 2007, 11.)

## 2.4.2 Ajax

Ajax on tämän hetken merkittävin vaihtoehto monipuolisten, dynaamisten verkkosovellusten luomiseen. Siinä ei varsinaisesti kuitenkaan ole mitään uutta. Ajaxin uusinkin tekniikka XMLHttpRequest (XHR) julkaistiin alun perin ActiveX-komponenttina vuonna 1999 Internet Explorer 5:n mukana. Web-selaimet ovat toki kehittyneet huomattavasti tämän jälkeen, ja muut selaimet ovat toteuttaneet XMLHttpRequest-rajapinnan ilman ActiveX:ää. (Asleson & Schutta 2007, 14.)

Jesse James Garret lanseerasi Ajax-käsitteen vuonna 2005. Alkujaan akronyymi AJAX syntyi yhdistelmästä *Asynchronous JavaScript and XML*, eli epäsynkroninen JavaScript ja XML. Nykyisin termiä käytetään kuvaamaan yleisemmin kaikkia niitä tekniikoita, joiden avulla selaimen on mahdollista hakea tietoa palvelimelta lataamatta koko sivua uudelleen. Juuri tätä tarkoitetaan sanalla epäsynkroninen. JavaScript on edelleen keskeinen osa Ajaxia, mutta senkin voi korvata esimerkiksi VBScriptillä, Rubylla tai Pythonilla. XML:n käyttö ei sekään ole pakollista. (Asleson & Schutta 2007, 17; Krill 2008; Salo 2010, 59.)

Kun koko sivua ei tarvitse jokaisen muutoksen tai tapahtuman yhteydessä ladata uudelleen, käytettävyys ja vuorovaikutteisuus paranevat. Ajaxin myötä nykyaikaiset verkkopalvelut ovat saaneet kaipaamaansa nopeutta. (Salo 2010, 59.)

JavaScriptin maine kehittäjien keskuudessa ei ole ollut paras mahdollinen, mutta Ajaxiin kohdistuvan mielenkiinnon ansiosta JavaScript-ohjelmointi on kuitenkin helpottunut nopeasti kehittyvien sovellus- ja testauskehysten avulla. (Asleson & Schutta 2007, 14.)

Ajaxin uusi lähestymistapa on merkittävä askel internetin toimintamallille, joka aikaisemmin rakentui yksittäisten pyyntöjen ja vastausten varaan. Web-sovellusten kehittäjät voivat nyt vapaasti suunnitella sovelluksensa toimimaan epäsynkronisesti. Tämä mahdollistaa monien aikaisemmin vain perinteisemmissä ohjelmistoissa onnistuneiden toimintojen toteuttamisen web-sovelluksessa. (Asleson & Schutta 2007, 14.)

### 3 MITÄ WEB 2.0 TARKOITTAAN?

Web 2.0:n voidaan ajatella koostuvan kolmesta keskeisestä, osittain toisiinsa su-lautuvasta elementistä. Näitä ovat rikkaat internetsovellukset, palvelukeskeinen arkkitehtuuri ja sosiaalinen verkko. Aito Web 2.0 -palvelu hyödyntää näitä kaikkia. Tällaisia palveluita on maailmassa lukematon määrä (Kuvio 5). Tässä pääluvussa käsitellään näitä kolmea elementtiä, kuten myös itse termin ”Web 2.0” osakseen saamaa kritiikkiä.



KUVIO 5. Kokoelma Web 2.0 -sivustojen logoja. (Batac 2009.)

### 3.1 Rikkaat internetsovellukset

Rikkaat internetsovellukset (*RIA, Rich Internet Applications*) ovat webissä toimivia sovelluksia, joissa suurin osa prosessoinnista tapahtuu käyttäjän omassa selaimessa. Selain ja palvelin ovat yhteydessä epäsynkronisesti. (Poikonen 2007, 2.)

Vanhanaikaisten, synkronisten internetsovellusten monimutkaistuessa niissä alkoi ilmetä käytettävyysongelmia. Sovellukset reagoivat käyttäjän toimiin hitaasti, heikentäen käyttökokemuksen laatua. Lisäksi ne lisäsivät räsistä palveluntarjoajan palvelimella, eivätkä yksinkertaisesti vain toimineet, jos verkkoyhteys katkesi. (Bozzon, Comai, Fraternali & Carughi 2006, 353.)

Rikkaat internetsovellukset pyrkivät ratkaisemaan nämä ongelmat. Ne pystyvät esittämään kehittyneemmän käyttöliittymänsä avulla monimutkaisempia prosesseja, sekä minimoimaan käyttäjän ja palvelimen välisen datan siirron. Ne myös siirtävät tiedon esitystavan hallinnan palvelimelta käyttäjän selaimelle. (Bozzon, Comai, Fraternali & Carughi 2006, 353.)

**LucidChart.** Joitakin tämän opinnäytetyön kaavioista on luotu LucidChart-nimisellä web-sovelluksella. LucidChartin perustoiminnot ovat ilmaisia ja se on varsin helppokäyttöinen ja nopea. Sen avulla useampi käyttäjä voi myös muokata samaa kaaviota tai diagrammia eri tietokoneilla, ja muutokset päivittyvät kaikille reaaliajassa.

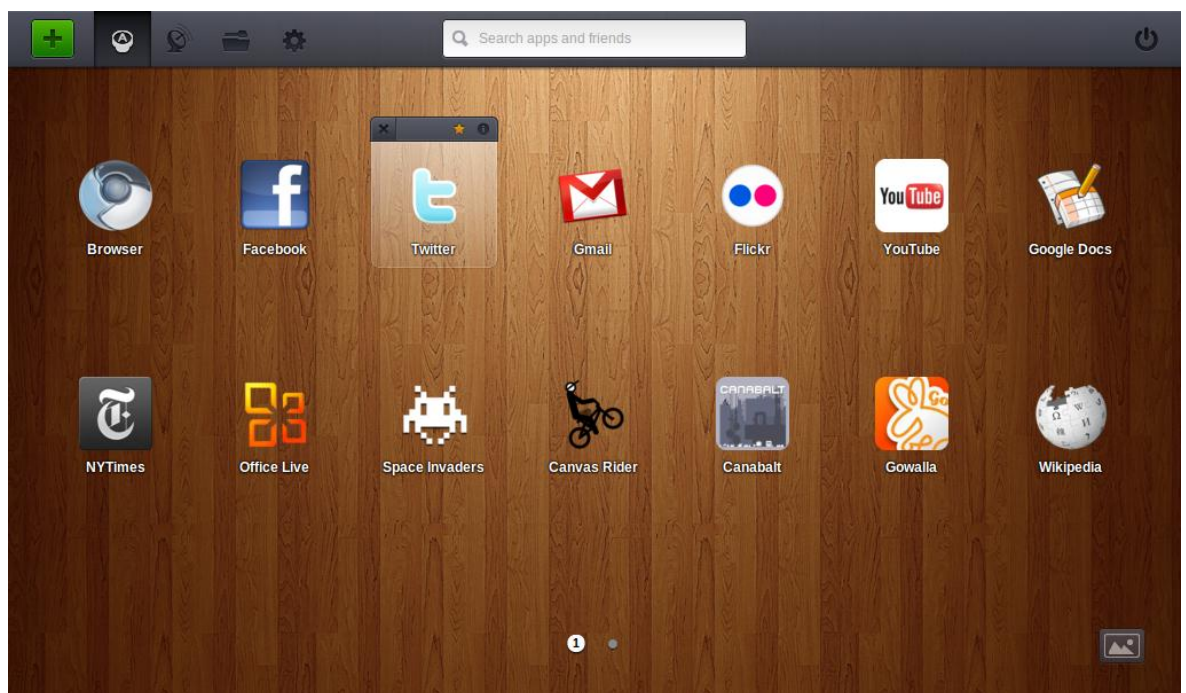
LucidChartia on verrattu perustoimintojensa osalta Microsoft Visioon. Vision edullisin hinta tätä kirjoittaessa kuitenkin oli 154,94 euroa. Suomenkieliset oikolukutoiminnot LucidChartista vielä puuttuvat, mutta hinta-laatu-vertailussa voittaja on kuitenkin selvä. Tämä on oiva esimerkki internetsovellusten mahdollisuuksista. (LucidChart 2011; Vertaa.fi 2011.)

**Web-käyttöjärjestelmät.** Googlen tuleva Chrome OS -käyttöjärjestelmä ei tule käyttämään natiiveja, erityisesti tietyille käyttöjärjestelmälle ohjelmoituja sovelluksia. Järjestelmän ydintoimintoja lukuun ottamatta kaikki sovellukset ovat internet-



pohjaisia. Internetsovellusten mahdollisuuksiin siis luotetaan, mutta aika näyttää, pystyvätkö internetsovellukset korvaamaan perinteiset sovellukset. (Google 2009.)

Myös jo julkaistun, Ubuntu-pohjaisen Joli OS:n (aikaisemmin tunnettu nimellä Jolicloud) käyttökokemus perustuu internetsovellusten käyttöön (Softpedia 2011). Sovellusvalikoimaa on kuitenkin mahdollista täydentää valikoiduilla natiivisti asennettavilla sovelluksilla. Näin saadaan käyttöön kummankin sovellustyyppin parhaat puolet. Web-sovellusten ja natiivien sovellusten välille ei käyttöliittymässä tehdä minkäänlaista eroa (Kuvio 6).



KUVIO 6. Joli OS:n työpöytä näkymä. (Jolicloud 2011.)

### 3.2 Palvelukeskeinen arkkitehtuuri

Palvelukeskeinen arkkitehtuuri (SOA, *Service Oriented Architecture*) on teknologiarippumaton suunnittelumalli, jonka perustan muodostavat itsenäiset liiketoimintalogiikkaa sisältävät palvelut. Nämä palvelut liitetään toisiinsa löyhän kytköksen (*loose coupling*) periaatteen mukaisesti, mikä vähentää järjestelmäriippuvuutta. Palvelukeskeinen arkkitehtuuri saa nämä järjestelmät keskustelemaan keskenään

yhteisen rajapinnan kautta, tuoden tiedon sitä tarvitsevien saataville ja helpottaen järjestelmien ylläpitoa. Palvelujen yhteistoiminnan mahdollistaa palveluväyläksi kutsuttu infrastruktuuri. (Siikaniemi 2008, 2–6, 10.)

Palvelukeskeistä arkkitehtuuria ei voi ostaa valmiina järjestelmänä, vaan se on pikemminkin strategiamalli ja suunnittelutapa. Suurimpia tekijöitä palvelukeskeisen arkkitehtuurin yhä lisääntyvään käyttöönnottoon ovat halu karsia IT-kustannuksia, parantaa organisaation ketteryttä ja liittää sisäisiä toimintoja paremmin yhteen. (Siikaniemi 2008, 2–5.)

### 3.2.1 Pilvipalvelut

Salon (2010, 16) mukaan konsulttiyhtiö Accenturen määritelmä pilvipalveluille (*Cloud Services*) on seuraavanlainen:

Pilvipalvelut ovat palveluntarjoajan IT-resurssien, kuten ohjelmistojen, laitteiston tai palvelujen, dynaamista tarjoamista asiakkaiden käyttöön verkon välityksellä. (Salo 2010, 16.)

Tämä tarkoittaa käytännössä sitä, että käyttäjä voi esimerkiksi tallentaa omia tiedostojaan palveluun käyttäjätunnustensa taakse. Tiedostoja voi sitten käyttää tai muokata miltä tahansa tietokoneelta pelkän web-selaimen tai mahdollisesti asennettavan asiakasohjelmiston avulla. Pilvipalvelut voivat johonkin pisteeseen asti korvata yritysten omat ohjelmistot ja laitteet. Kyse on siis tiedonvälityksen, laskenta- ja tallennuskapasiteetin tarjoamisesta käyttäjille internetin välityksellä.

Pilvipalvelumallissa tietotekniikkaresursseja tarjotaan käyttöön ilman, että käyttäjän tarvitsee tietää, missä resurssit fyysisesti sijaitsevat, tai huolehtia niiden toiminnasta tai ylläpidosta. (Salo 2010, 16.)

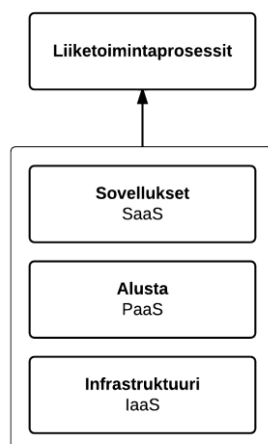
Pilvipalvelut ovat useimmiten melko edullisia. Yleinen toimintamalli on antaa käyttäjän käyttää osaa palvelusta ilmaiseksi ja periä edistyneempien toimintojen tai suuremman levytilan käytöstä kuukausimaksua.

Pilvipalvelut voidaan jakaa kolmeen liiketoimintaprosesseja tukevaan kerrokseen (Kuvio 7):

**Infrastruktuuri palveluna (*IaaS, Infrastructure as a Service*).** Asiakas ostaa palveluntarjoajan resurssit palveluna käyttöönsä. Tarjottu kapasiteetti on usein virtualisoitu ja se skaalautuu asiakkaan tarpeisiin. Laskutus perustuu käytettyihin resursseihin. (Salo 2010, 25.)

**Sovellusalusta palveluna (*PaaS, Platform as a Service*).** Asiakas ostaa palveluna alustan, jonka avulla voi rakentaa, kehittää ja ylläpitää sovelluksia. Hyötyihin kuuluvat muun muassa kehitystyön nopeutuminen ja suurempi kustannustehokkuus, kuten myös mahdollisuus käyttää sovelluksissa valmiita moduuleja ja lisäosia. Haittapuoliksi taas lasketaan sitoutuminen tiettyyn palveluntarjoajaan, uudet osaamisvaatimukset kehitystyössä sekä tietoturvahuolet. (Salo 2010, 28–29.)

**Sovellukset palveluna (*SaaS, Software as a Service*).** Asiakas ostaa käyttöönsä sovellukset palveluna, eikä itse joudu ylläpitämään tai päivittämään niitä. Perinteisen ohjelmistolisenssimaksun sijaan asiakas maksaa esimerkiksi aikaperusteisen käyttäjä- tai laitekohtaisen maksun. SaaS-palvelun mahdollistamat resurssit ovat tehokkaassa käytössä sovellusarkkitehtuurissa, joka tukee useaa yhtäaikaista käyttäjää. Noin puolet pilvipalveluiden arvosta muodostuu SaaS-palveluista. (Salo 2010, 22, 29.)



KUVIO 7. Pilvipalvelut liiketoimintaprosessien tukena (Salo 2010, 23)

Pilvipalvelumalleja voidaan jakaa ryhmiin myös muilla tavoilla. Omiksi kokonaisuuksikseen voidaan laskea esimerkiksi tallennustila palveluna (*Storage as a Service*), tietoturva palveluna (*Security as a Service*) tai viestintä palveluna (*Communication as a Service*). (Salo 2010, 23.)

Esimerkkeinä suosituista SaaS-pilvipalveluista, joita pystyy käyttämään pelkkänä web-sovelluksena ilman ylimääräisten sovellusten asentamista, voidaan mainita seuraavat:

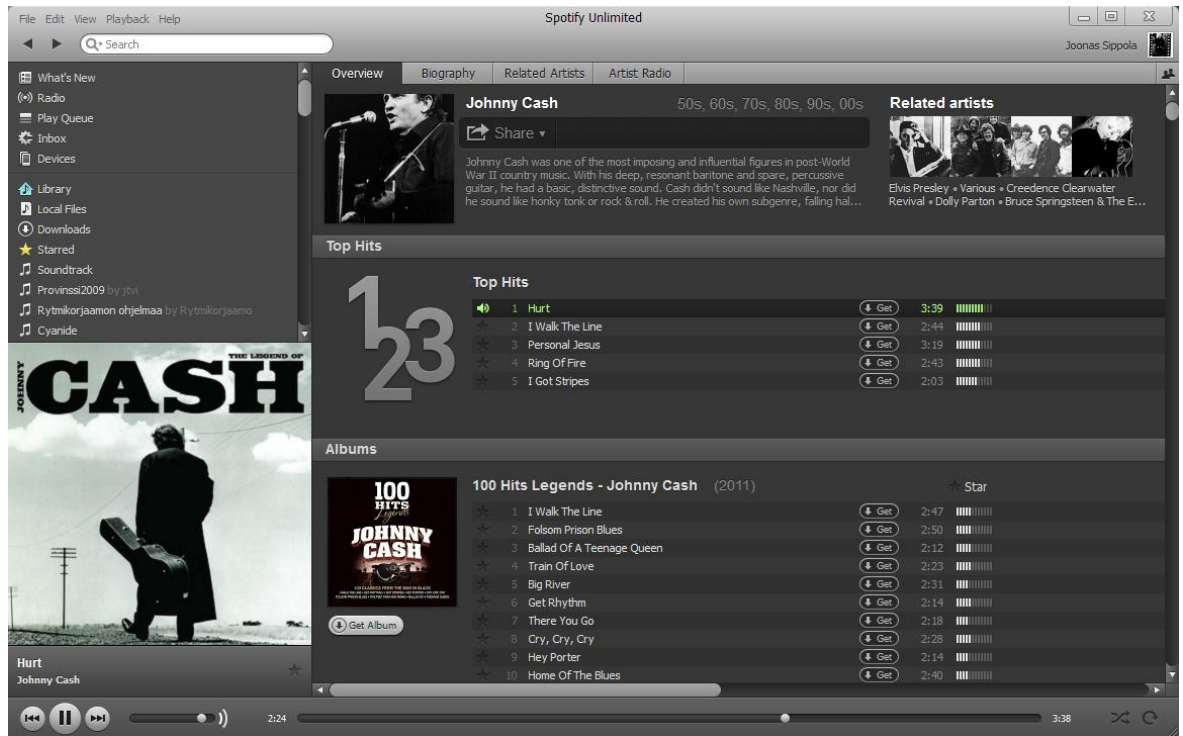
- **Flickr** antaa käyttäjien tallentaa ja jakaa valokuvia muiden käyttäjien kanssa
- **Youtube** toimii samalla periaatteella, mutta jakaa videoita
- **Gmail** tarjoaa käyttäjilleen epäsynkronisen ja joustavan sähköpostisovelluksen ja tätä kirjoittaessa 7,5 gigatavua tallennustilaa sähköposteille. Gmail mahdollistaa myös reaaliaikaisen keskustelun käyttäjien välille.
- **Google Docs** mahdollistaa käyttäjilleen tiedostojen muokkaamisen kehittyneillä toimistotyökaluilla (myös Microsoft kehittää omaa pilvi-toimisto-ohjelmistoaan).
- **Dropbox** tarjoaa käyttäjilleen tallennustilaa internetissä (asennettava asiakasohjelmisto synkronisoi tiedostoihin tehdyt muutokset kaikille käyttäjän laitteille).

(Salo 2010, 155–156.)

Seuraavat suositut SaaS-palvelut taas vaativat asiakasohjelmiston asentamisen:

- **Vodderin** käyttäjä voi katsoa vuokraelokuvia virtautustekniikalla (*streaming*), lataamatta koko elokuvatiedostoa ensin omalle tietokoneelleen.
- **Spotifyn** avulla käyttäjä voi kuunnella musiikkia virtautustekniikalla suoraan internetistä Vodderin tavoin (Kuvio 8).
- **Panda Cloud Antivirus** ei lataa päivityksiä eikä rasita käyttäjän tietokoneen prosessoria tarkastuksissa, vaan tarkistaa uhkaavat tiedostot keskittelyllä palvelimella.

(Salo 2010, 155–156.)



KUVIO 8. Spotifyn asiakasohjelmisto.

**Riskit.** Pilvipalveluiden käyttämisessä on kuitenkin riskinsä. Suosittua IaaS-palvelua tarjoava AWS EC2 (Amazon Web Services Elastic Cloud Compute) tais-teli teknisiä ongelmia vastaan 21–22. huhtikuuta 2011, ja tänä aikana monet sen palveluista riippuvaiset yritykset eivät pystyneet palvelemaan asiakkaitaan. Talou-dellisen vahingon lisäksi yritysten luotettavuus kuluttajan näkökulmasta saattoi vahingoittua. Merkittävä huoli on myös tietoturva, koska yrityksen käyttämät sovel-lukset ja niihin liittyvä data saattavat sijaita yrityksen palomuurin ulkopuolella pal-veluntarjoajan tiloissa. Luottamus palveluntarjoajan kykyyn suojautua tietomurtoja vastaan täytyy olla suuri. (Salo 2010, 29; Lohr 2011.)

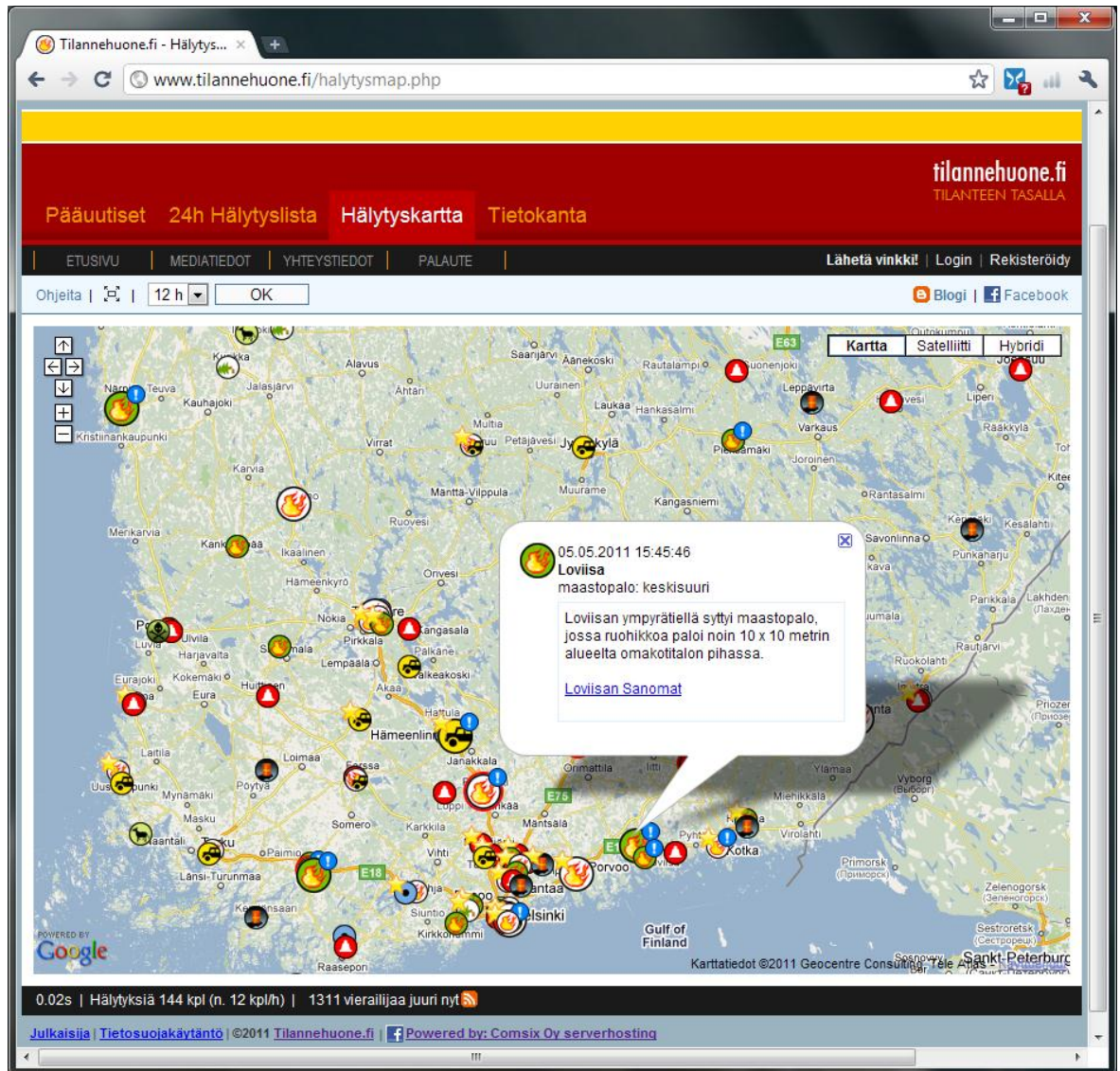
### 3.2.2 Yhdistelmäpalvelut

Yhdistelmäpalvelut (*Mashups*) tarkoittavat erilaisten tietokantojen ja -lähteiden yhdistelyä ja tehokkaampaa hyödyntämistä. Ne perustuvat yleensä ulkopuolisten, maksuttomien ja helposti saatavilla olevien käyttöliittymien ja tietojen käyttämiseen. Yhdistelmäpalveluiden laatiminen ei ole erityisen vaikeaa ja se onkin ollut viime aikoina melko suosittua. (Tirronen 2008, 77.)

Esimerkiksi Google, Microsoft ja Yahoo sekä useat muut suuret verkkotoimijat tarjoavat Web API:n, eli ohjelmoitavan www-palvelun rajapinnan. Yhdistelmäpalvelu liittää usean eri lähteen rajapinnat yhteyteen toistensa kanssa, jolloin niiden tietoja on mahdollista soveltaa ristikkäin. Google Maps -karttapalvelu on yleisesti ollut hyvin yleinen yhdistelmäpalvelun komponentti. Karttojen yhteyteen on mahdollista yhdistää monenlaisia tietoja, ja kartat voivat selkeyttää monimutkaisen tiedon esittämistä. (Tirronen 2008, 77.)

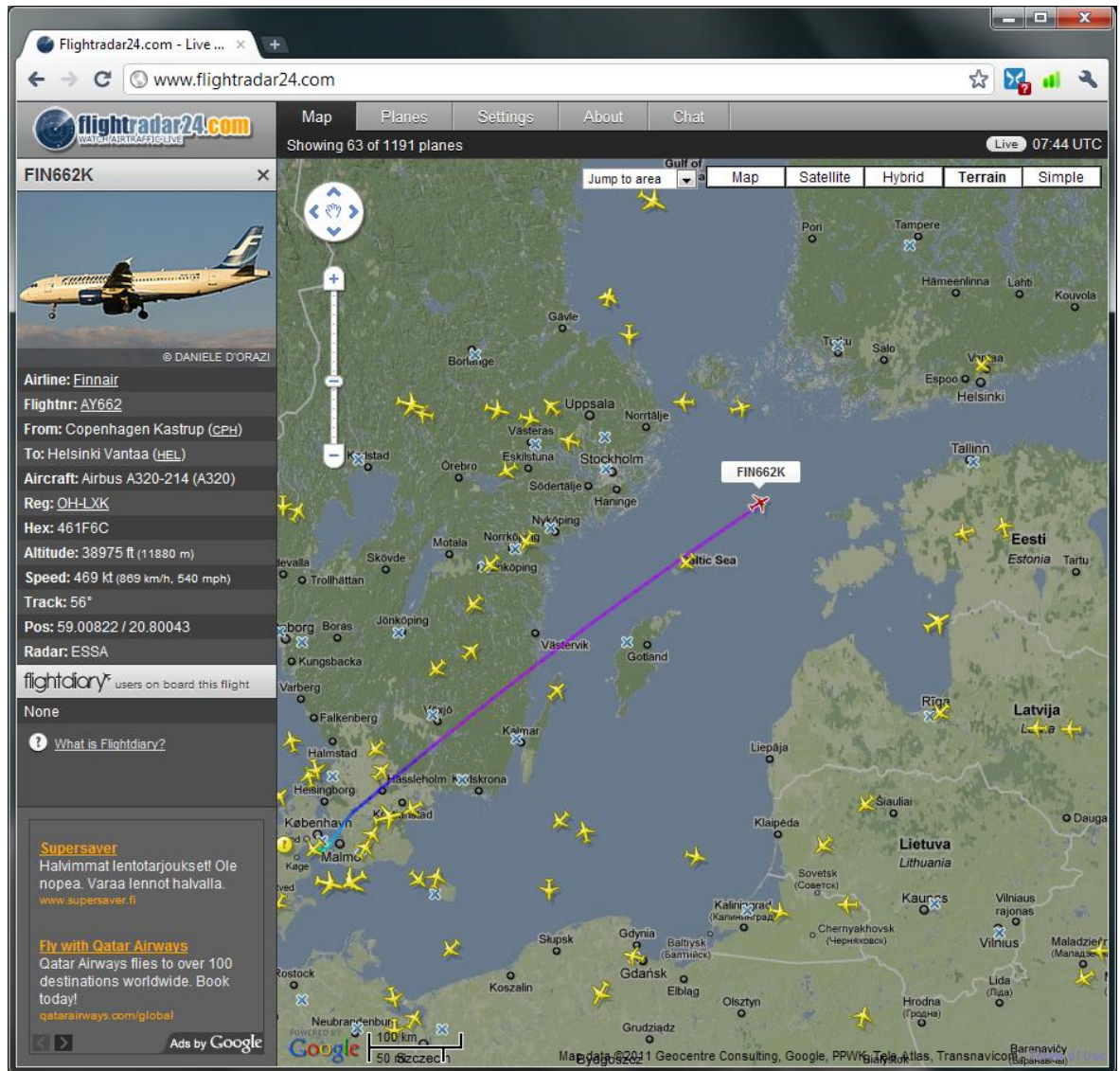
Yhdysvalloissa avoin, eli vapaasti saatavilla oleva data on johtanut yhdistelmäpalveluiden määrän voimakkaaseen nousuun. Avoin data myös tarjoaa ohjelmoijille monia mahdollisuuksia. Avoimen datan eteen on tehty Suomessakin paljon töitä vuosina 2009–2010. (Hintikka 2011.)

Suomalainen Tilannehuone.fi esittää tuoreimmat hälytystiedot Google Mapsin avulla. Tiedot saadaan 112infon ja Pronon julkisista listoista (Tirronen 2008, 77). Palvelu sijoittaa Suomen kartan päälle symboleja, joista selviää hälytyksen tyyppi ja vakavuus. Valitsemalla symbolin on mahdollista saada hälytyksestä lisätietoja, kuten hälytysajan sekä mahdollisesti kuvauksen ja valokuvia paikan päältä (Kuvio 9).



KUVIO 9. Tilannehuone.fi:n hälytyskartta. (Tilannehuone.fi 2011.)

Toinen mielenkiintoinen, niin ikään Googlen kartoja käyttävä yhdistelmäpalvelu on Flightradar24.com. Se piirtää kartan päälle reittilentojen tietoja (Kuvio 10). Palvelun avulla käyttäjän on mahdollista esimerkiksi katsoa minkä näköinen lentokone ylitti juuri hänen kotikaupunkinsa. Palvelusta selviää myös koneen lentokorkeus, -nopeus ja määränpää. Vaikka näiden kahden palvelun perusidea on samankaltainen, niiden visuaalinen toteutus on kuitenkin varsin erilainen.



KUVIO 10. Flightradar24.com:n reittilentokartta. (Flightradar24.com 2011.)

### 3.3 Sosiaalinen verkko

Sosiaalisuus on tärkeä osa Web 2.0:aa. Yksi sen ilmenemismuotoja on sosiaalinen media. Sosiaalisella mediallyä tarkoitetaan internetin käyttäjien itsensä luomaa ja toisilleen jakamaa sisältöä. Yksinkertaisimmillaan se voi tarkoittaa vaikkapa vain blogia, linkkikokoelmaa tai omien valokuvien lisäämistä kuvien tallennus- tai jakopalveluun. Riippumatta siitä, millaisesta sisällöstä on kyse, keskeistä on jakaminen (Tirronen 2008, 34). Erilaisia sisällönjakamiseen keskittyviä palveluita on monenlaisia (Kuvio 11).





Tärkein tekijä blogikulttuurin taustalla on ollut blogien julkaisemisen helppous, nopeus ja yksinkertaisuus. Myös monet menestyneet ja hyvin kannattavat yhtiöt käyttävät blogeja monella tavalla sekä sisäisessä että ulospäin suuntautuvassa viestinnässään. (Kilpi 2006, 3.)

Blogit edustavat Tirrosen (2008, 55) mukaan ruohonjuuritasolla kaikkea sitä, mistä Web 2.0:ssa on pohjimmiltaan kyse:

Ne rakentuvat kaikista niistä ajattelumalleista, toimintatavoista ja teknisistä ratkaisuista, jotka muodostavat Web 2.0 -maailman ytimen. Blogeissa yhdistyvät yhteisöllisyys, sosiaalisuus, dynaamisuus, sisällön tuottaminen ja jakaminen sekä verkottuneisuuden voima. (Tirronen 2008, 55.)

**Nettipäiväkirja-termi.** Blogeja kutsutaan joskus nettipäiväkirjoiksi. Useat bloggajat eivät hyväksy tätä termiä, koska he mieltävät päiväkirjaa varten tuotetun kirjoituksen pikemmin itsekeskeiseksi ja sisäänpäin kääntyneeksi. Nykysuomen Sanakirja kuitenkin määrittelee Kilven (2006, 5) mukaan päiväkirjan olevan ”päivittäin tai tarpeen mukaan tehdyt muistiinpanot tapauksista, kokemuksista, joko kronikan mukaan tai kaunokirjallisesti esitettyinä”. Tämä määritelmä sopii hyvin myös useimpiin blogeihin. (Kilpi 2006, 5.)

**Blogien kehitys.** Varhaiset blogit olivat käsin HTML-kielellä kirjoitettuja verkkosivuja. Verkossa julkaiseminen oli mahdollista vain niille, jotka osasivat itse tuottaa verkkosivuja. Erilaiset blogialustat ja -palvelut muuttivat tilanteen 2000-luvulla ja tekivät verkkojulkaisemisen yhtä helpoksi kuin tavallisen tekstinkäsittelyohjelman käyttämisen. (Kilpi 2006, 4.)

### 3.3.2 Sosiaaliset verkostot

Sosiaaliset verkostot, kuten Facebook tai LinkedIn, ovat hyvä esimerkki Web 2.0:n mahdollisuuksista. Verkostot toimivat ihmisten välisen vuorovaikutuksen kanavina ja alustoina. Facebookin yli 500 miljoonan käyttäjän nopeasti kasvava massa tekee siitä myös houkuttelevan PaaS-sovellusalustan miljoonille sovelluskehittäjille. (Salo 2010, 38–39.)

### 3.4 Web 2.0 -käsitteen tarpeellisuudesta herännyt kritiikki

World Wide Webin kehittäjä Sir Tim Berners-Lee on kutsunut Web 2.0 -termiä siansaksaksi. Se ei hänen mukaansa tuo määritelmään mitään uutta. Kaikki Web 2.0:n määrittelemät ominaisuudet ovat Lanninghamin (2006) mukaan alusta asti kuuluneet Berners-Leen alkuperäiseen visioon:

Joillekin ihmisille Web 2.0 tarkoittaa osan ajatustyöstä siirtämistä selaimen puolelle tehden siitä välittömämpää, mutta ajatus webistä ihmisten välisenä vuorovaikutuksena on webin ydin. Sitä varten se tarkoitettiin, yhteiseksi tilaksi, jossa ihmiset voivat vuorovaikuttaa. (Lanningham 2006.)

Web 2.0 -käsitteen arvostelijat huomauttavat jatkuvasti sosiaalisen vuorovaikutuksen ja jakamisen kulttuurin olleen koko ajan webin alkuperäisenä tavoitteena. Yhteisöllisyys ei kuitenkaan koskaan aiemmin ole toteutunut näin laajasti. Vaikka osa Web 2.0 -termin toistamisesta onkin palveluiden rakentajille vain hyvää markkinointia, kyseessä on kuitenkin aito ilmiö, joka vasta nyt on saavuttanut massojen suosion. Nykyisissä laajoissa yhteisöllisissä palveluissa alkuperäinen ajatus yhteisöllisyydestä on kypsynyt luonnollisesti ja on saavuttamassa alkuperäistä potentiaaliansa. (Tirronen 2008, 77.)

Vaikka Web 2.0 terminä ei välttämättä olekaan parhain mahdollinen, se kuvaa silti melko hyvin nykypäivän verkkoa ja sen kehitysprosesseja, olettaen kuitenkin, että versionumerolta malttaa olla vaatimatta liiallista tarkkarajaisuutta ja selväpiirteisyyttä. (Tirronen 2008, 34.)

**Web 2.0 muoti-ilmiönä.** Web 2.0 -sivustojen nimissä on havaittavissa selkeä trendi. Tämän trendin puolesta puhuvat muun muassa nimet Tumblr, Flickr, Socializr, Twitter (alun perin Twtr), Shyftr, Jpgr, Kontactr, Bloglinkr, Wishlistr, Viewr, DiggFiltr, Rsizr ja Readr (Weaver 2006; Solis 2008; Batac 2009). Nimet kuvastavat toiminnallisuutta. Sivustot eivät halua tulla nähtäväksi vain passiivisina kohteina, vaan pikemminkin aktiivisina toimijoina. Nimien loppuosien muodikkaan lyhentämisen taas voi ajatella olevan samantapainen ilmiö kuin 1990-luvulla viljeltyt ylimääräiset X- ja Z-kirjaimet.

## 4 TULEVAISUUS

Tässä pääluvussa luodaan silmäys lähitulevaisuuteen ja esitellään arvioita, minkälaiseksi internet saattaa tulevaisuudessa kehittyä.

### 4.1 HTML5

HTML5 on W3C:n uusin, tätä kirjoittaessa vielä luonnosvaiheessa oleva HTML-suositus. Vaikka sen määritelmää ei ole virallisesti vahvistettu, useimmat elementit ovat jo paikoillaan. HTML5:n ominaisuuksia hyödyntävien verkkosivujen määrä kasvaa jatkuvasti. (W3C 2011a.)

HTML5 tukee monia uusia menetelmiä, esimerkkeinä mainittakoon tiedostojen raahaaminen työpöydältä selaimen ja päinvastoin, dynaamiset animaatiot, audio- ja videosisällön upottaminen sivulle, kolmiulotteisen grafiikan piirtäminen, tekstin varjot ja heijastukset sekä liukuvärjäykset (Google 2011). HTML5:ttä on muutenkin pyritty markkinoimaan tulevaisuuden menetelmänä, ja W3C on julkaissut sille jopa logon (Kuvio 12).

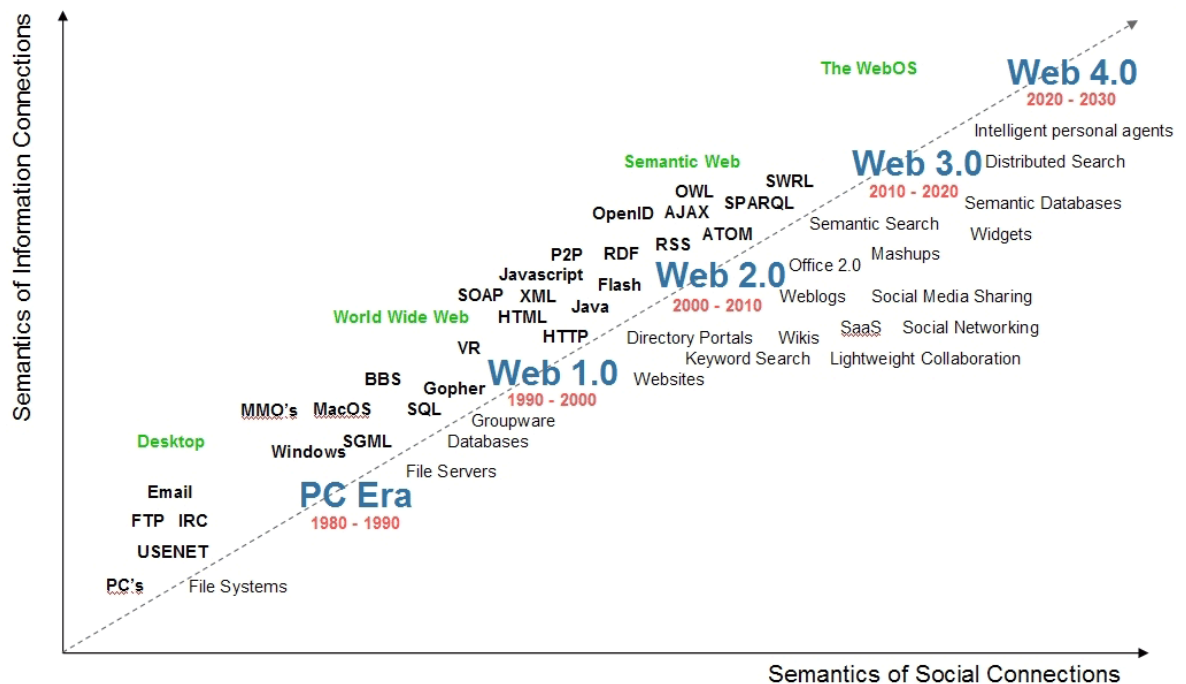


KUVIO 12. HTML5:n logo. (W3C 2011b.)

## 4.2 Web 2.0:n jälkeen

**Web 3.0.** Internetistä on ajateltu Web 3.0:n myötä tulevan semanttinen verkko, joka ymmärtäisi tiedon merkityssuhteet. Tällainen verkko osaisi esimerkiksi antaa järkevän vastauksen kysymykseen ”Etsin lämmintä lomakohdetta ja minulla on 3 000 euron budjetti. Tämän lisäksi minulla on myös 11-vuotias lapsi.” (Talouselämä 2006). Web 3.0:n määritelmä on kuitenkin vielä epäselvä, ja semanttisia sovelluksia ei välttämättä ole luvassa vielä 10 vuoteen. (Farber 2007.)

**Web 4.0.** Web 4.0:n määritelmästä ei tässä vaiheessa vielä voi oikein edes puhua. Vakuuttavimman kuuloisetkin visiot ovat kuitenkin vain arvailua (Spivack 2007). Joidenkin näistä veikkauksista mukaan verkko olisi 25 vuoden päästä jo aidosti älykäs, ja sen kanssa voisi keskustella kuin ihmisen kanssa. Spivackin (2007) mukaan älykkäät henkilökohtaiset toimijat alkaisivat yleistyä jo ensi vuosikymmenen alkupuolella (Kuvio 13). (Farber 2007; PCWorld 2008.)



KUVIO 13. Kaavio kehityksestä kohti Web 4.0:aa. (Spivack 2007.)

## 5 NÄYTESIVUJEN RAKENTAMINEN

Tässä pääluvussa kerrotaan näytesivujen rakentamisesta. Tavoitteena on suunnitella ja rakentaa näytesivuja, jotka esittelisivät joitain Web 2.0:n tärkeimpiä ominaisuuksia.

### 5.1 Suunnittelu

Ensimmäinen näytesivu haluttiin toteuttaa PHP-sovelluskehityksen avulla, koska opinnäytetyön tekijällä ei ollut aikaisempaa kokemusta PHP:n käyttämisestä kehysten avulla. Mahdollisista sovelluskehityksistä CodeIgniter vaikutti mielenkiintoisimmalta. CodeIgniterin vakuuttavan näköiset oppaat myös helpottivat valintaa.

Toiseen näytesivuun haluttiin tuoda Ajax-toiminnallisuutta. Tähän parhaalta vaihtoehdolta vaikuttivat JavaScript-kirjaston tarjoamat valmiit toteutukset. Käytettäväksi kirjastoksi valittiin suosittu jQuery. Toiseen näytesivuun päätettiin yhdistää myös sosiaalinen elementti Facebook-integroinnin muodossa.

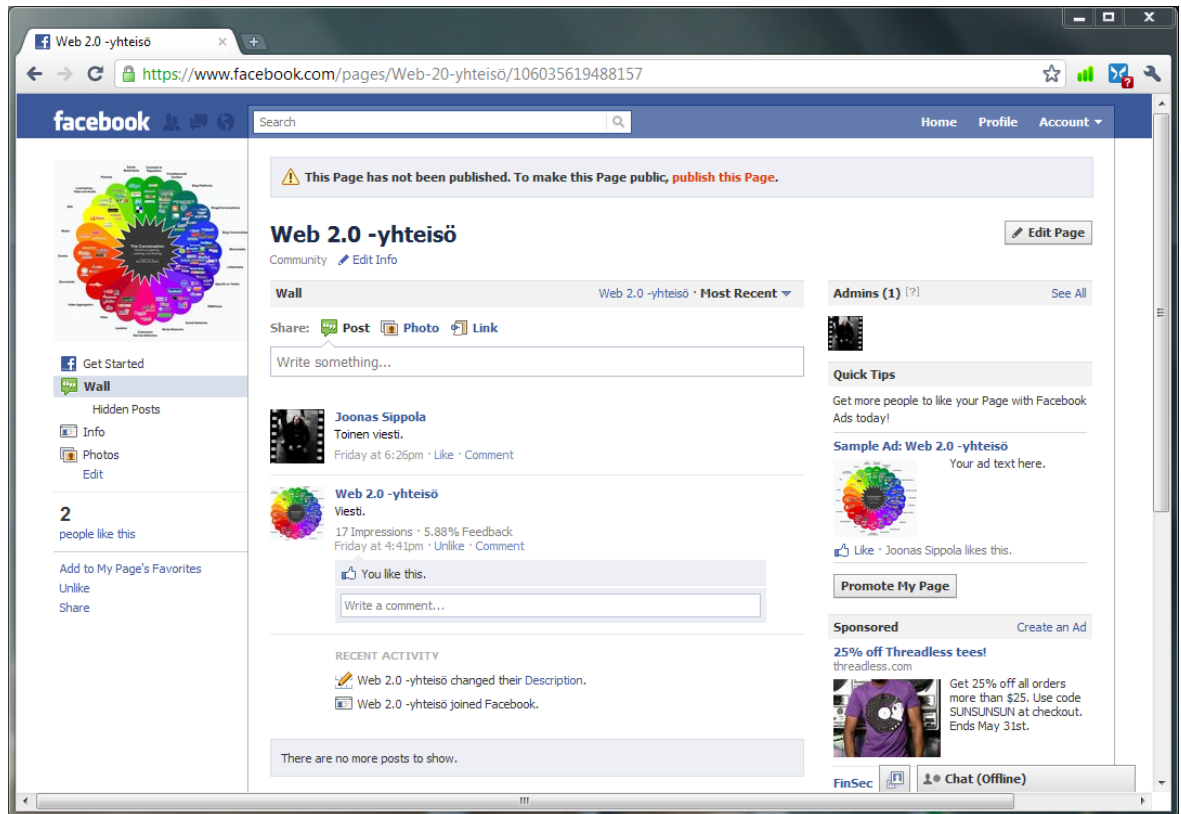
### 5.2 Toteutus

Ensimmäinen asia, mitä dynaamisten sivujen rakentamiseen tarvitaan, on palvelin. Tähän tehtävään päätettiin käyttää XAMPP-ohjelmistoa. XAMPP on kehityskäyttöön suunnattu ohjelmisto, johon on koottu palvelinkäytössä tarpeellisia sovelluksia, kuten Apache HTTP Server, MySQL-tietokantasovellus ja PHP-tulkki. XAMPP:n asennus oli yksinkertainen ja onnistui nopeasti.

**Blogi.** Ensimmäiseksi näytesivuksi päätettiin toteuttaa blogi. CodeIgniterin sivuilla oli ohjeet blogin perustoimintojen toteuttamiseen, mutta ne olivat vanhentuneet ja sisälsivät viittauksia käytöstä poistettuihin funktioihin. Tästä johtuen rakentaminen katkesi monta kertaa virhetilanteisiin. Näistä kuitenkin selvittiin korvaavien funktioiden nimet selvittämällä. Blogin kirjoitusten ja kommenttien tallentamiseen käytettiin MySQL-tietokantaa.

**Arvosanasovellus.** Toisen näytesivun Ajax-toiminnallisuuteen löydettiin malli jQueryn oppaista. Kyseessä oli sovellus, jonka avulla web-sivulle oli mahdollista toteuttaa arvostelutoiminto. Sovellus pyytää käyttäjää antamaan arvosanan, jonka vaihtoehtoisiksi tarjotaan linkkeinä numerot yhdestä viiteen. Kun käyttäjä on tehnyt valintansa, äänestyslinkit poistuvat sivulta ja niiden tilalle ilmestyy teksti, joka kertoo äänten keskiarvon ja määrän. Sovellus käyttää äänten tallentamiseen PHP-skriptiä, joka tallentaa ne ns. *flat file* -menetelmällä suoraan tiedostoon käyttämättä tietokantaa. Tämä kaikki toteutetaan epäsynkronisesti Ajaxilla, eikä koko sivua tarvitse missään vaiheessa ladata uudestaan.

Facebook tarjoaa web-kehittäjälle mahdollisuuden upottaa web-sivulle Facebookista tutun kommenttilaatikon. Tätä varten piti ensiksi luoda Facebook-sivu (Kuvio 14). Tämän jälkeen Facebook Developers -sivulta pystyi hakemaan HTML-koodin, jonka avulla upotus onnistui.



KUVIO 14. Luotu Facebook-sivu.

### 5.3 Tulokset

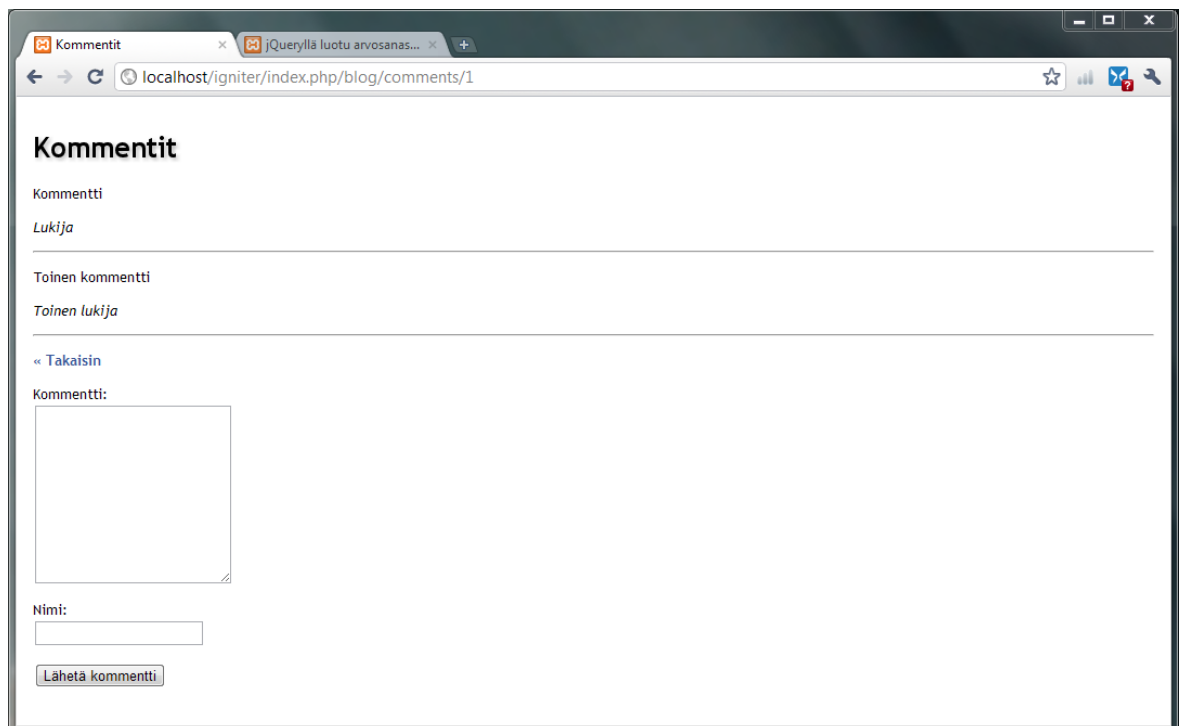
Lopputuloksena saatiin kaksi toimivaa dynaamista web-sivua, jotka esittelevät Web 2.0:n toimintaperiaatteita yksinkertaisimmillaan. Blogi hakee kirjoitukset tietokannasta ja näyttää ne lukijalle (Kuvio 15). Lukija voi halutessaan kommentoida kirjoituksia ja ne tallennetaan tietokantaan kommentoijan nimellä varustettuna (Kuvio 16). Arvosanasovellus tarjoaa arvosanavaihtoehdot (Kuvio 17), tallentaa valinnan ja esittää lopullisen keskiarvon lataamatta sivua uudelleen (Kuvio 18). Myös Facebook-komentointi toimii epäsynkronisesti.

**Johtopäätökset.** Jos aikaa olisi jäänyt enemmän, blogiin olisi myös voinut lisätä kirjoitusten päivämäärät ja syötteiden tarkistukset. Arvosanasovelluksen olisi voinut yhdistää blogikirjoituksiin ja muuttaa arvosanojen tallennuskohteen tiedoston sijaan tietokannaksi.

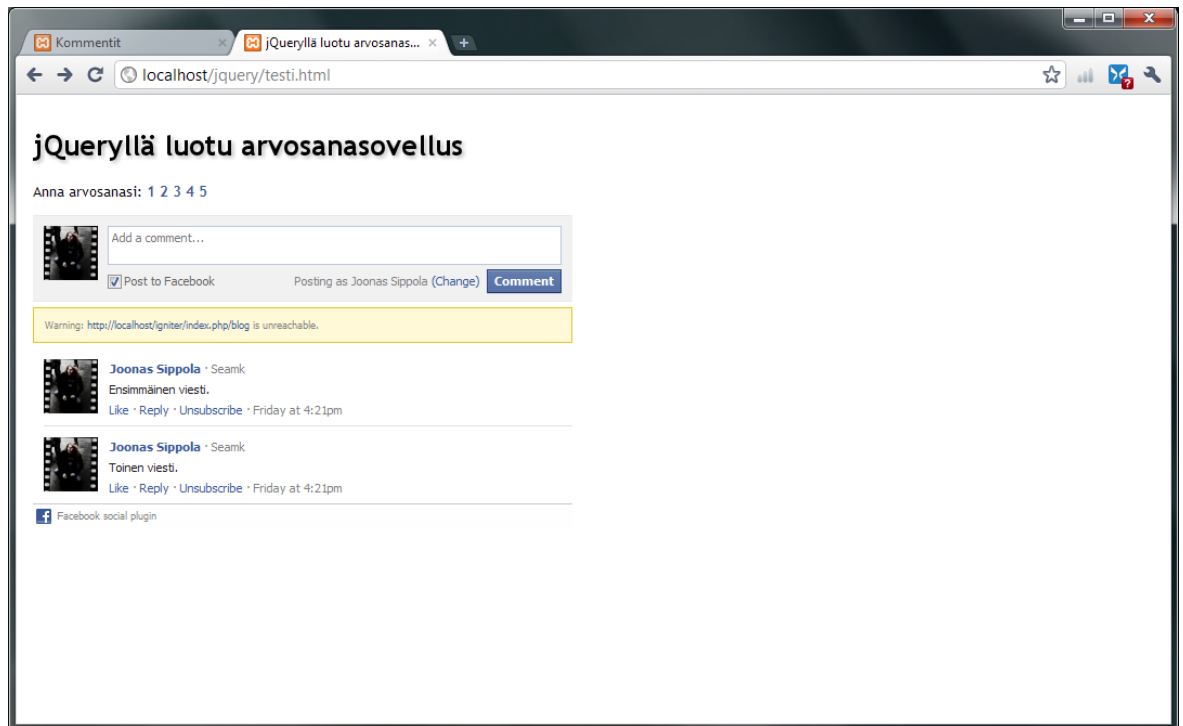




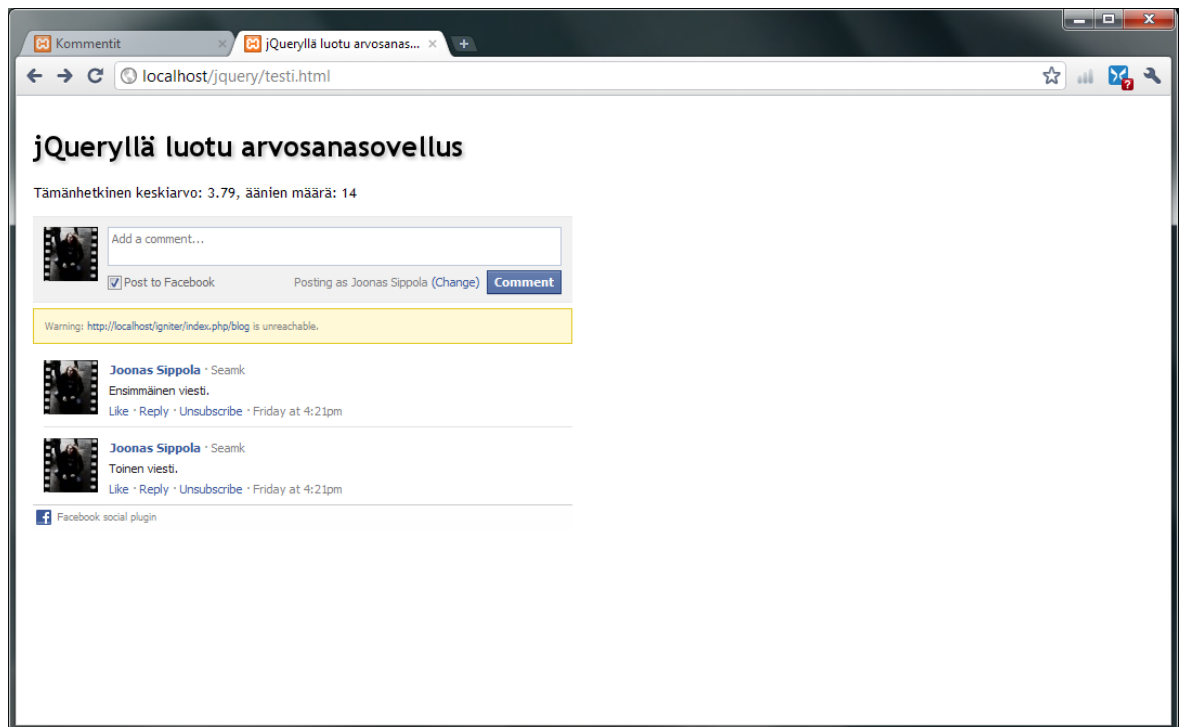
KUVIO 15. Blogin etusivu.



KUVIO 16. Blogin kommenttisivu.



KUVIO 17. Arvosanasovelluksen alkutila.



KUVIO 18. Arvosanasovellus esittää arvosanojen keskiarvon.

## 6 YHTEENVETO

Web 2.0 on uskomattoman laaja kokonaisuus. Mitä pidemmälle opinnäytetyön kirjoittaminen eteni, sitä selvemmäksi kävi, ettei ilmiötä voisi mitenkään käsitellä tyhjentävästi. Työssä pyrittiin mahdollisimman hyvin keskittymään oleellisimpiin asioihin, mutta monia alueita jäi varmasti silti esittelemättä.

Tekijälle hyödyllisin osa opinnäytetyön tekemistä oli tutustuminen PHP-sovellukehyksiin ja JavaScript-kirjastoihin. Näihin jatkossa lisää perehtymällä monimutkaisten web-sivustojen kehittämistä olisi mahdollista yksinkertaistaa ja helpottaa. Myös HTML5 on kiinnostava tekniikka, jota ilman web-kehittäjä ei enää tulevaisuudessa välttämättä pärjää.

## LÄHTEET

- Agarwal, R. 2010. 15 Best Ever PHP Frameworks. [www-dokumentti]. Admix Web. [Viitattu 31.3.2011]. Saatavissa: <http://www.admixweb.com/2010/10/06/15-best-ever-php-frameworks/>
- Asleson, R & Schutta N. T. 2007. Ajax – Tehokas hallinta. Helsinki: Readme.fi.
- Batac, A. J. 2009. World of Web 2.0. [www-dokumentti]. Soller Thoughts. [Viitattu 18.4.2011]. Saatavissa: <http://sollerthoughts.co.uk/2009/02/13/world-of-web-20/>
- Bozzon, A., Comai, S., Fraternali, P. & Carughi, G.T. 2006. Conceptual modeling and code generation for rich internet applications. Teoksessa: Proceedings of the 6th international conference on Web engineering. [Verkkokirja]. New York: ACM Press, 353-360. [Viitattu: 28.5.2011]. Saatavissa: <http://portal.acm.org/citation.cfm?id=1145649>
- BuiltWith Technology Usage Statistics. 2011. JQuery Usage Statistics. [www-dokumentti]. BuiltWith Technology Usage Statistics. [Viitattu 22.3.2011]. Saatavissa: <http://trends.builtwith.com/javascript/JQuery>
- Ecma International. 2009. [pdf-dokumentti]. ECMAScript Language Specification. [Viitattu 3.3.2011]. Saatavissa: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>
- Ek, J. & Norén K-J. 1999. Dynaaminen HTML käytännössä. Schildts Kustannus Oy - Pagina.
- Farber, D. 2007. From semantic Web (3.0) to the WebOS (4.0). [www-dokumentti]. ZDNet. [Viitattu 28.3.2011]. Saatavissa: <http://www.zdnet.com/blog/btl/from-semantic-web-30-to-the-webos-40/4499>
- Flanagan, D. 2006. JavaScript: The Definitive Guide. 5. p. [Verkkokirja]. O'Reilly Media, Inc. [Viitattu: 3.3.2011]. Saatavissa: <http://my.safaribooksonline.com/0596101996/jsript5-CHP-1?portal=oreilly#X2ludGVybmFsX0ZsYXNoUmVhZGVyP3htbGkP TA1OTYxMDE5OTYvMg==>

- Flightradar24.com. 2011. Flightradar24.com – Live Flight Tracker! [www-dokumentti]. Flightradar24.com. [Viitattu 29.3.2011]. Saatavissa: <http://www.flightradar24.com/>
- Foley, M. J. 2010. Microsoft: Our strategy with Silverlight has shifted. [www-dokumentti]. ZDNet. [Viitattu 2.4.2011]. Saatavissa: <http://www.zdnet.com/blog/microsoft/microsoft-our-strategy-with-silverlight-has-shifted/7834>
- Google. 2009. Introducing the Google Chrome OS. [www-dokumentti]. Google Inc. [Viitattu 29.3.2011]. Saatavissa: <http://googleblog.blogspot.com/2009/07/introducing-google-chrome-os.html>
- Google. 2011. HTML5 Presentation. [www-dokumentti]. Google Inc. [Viitattu 28.3.2011]. Saatavissa: <http://slides.html5rocks.com/>
- Hintikka, K. A. 2011. Yhteiskunta verkossa. Mikrobitti (3), 49.
- Jobs, S. 2010. Thoughts on Flash. [www-dokumentti]. Apple Inc. [Viitattu: 4.3.2011]. Saatavissa: <http://www.apple.com/hotnews/thoughts-on-flash/>
- Jolicloud. 2011. Welcome to the new Jolicloud! – Jolicloud Blog. [www-dokumentti]. Jolicloud. [Viitattu 8.5.2011.] Saatavissa: <http://www.jolicloud.com/blog/2011/03/09/welcome-to-the-new-jolicloud/>
- Kilpi, T. 2006. Blogit ja bloggaaminen. Helsinki: Readme.fi.
- Korpela J. K. & Linjama T. 2005. Web-suunnittelu. Jyväskylä: Docendo Finland Oy.
- Krill, P. 2008. JavaScript creator ponders past, future. [www-dokumentti]. InfoWorld. [Viitattu 22.3.2011]. Saatavissa: <http://www.infoworld.com/d/developer-world/javascript-creator-ponders-past-future-704>
- Laningham, S. 2006. developerWorks Interviews: Tim Berners-Lee. [www-dokumentti]. IBM developerWorks. [Viitattu 19.2.2011]. Saatavissa: <http://www.ibm.com/developerworks/podcast/dwi/cm-int082206txt.html>
- Lohr, S. 2011. Amazon's Trouble Raises Cloud Computing Doubts. [www-dokumentti]. The New York Times Company. [Viitattu: 25.4.2011]. Saatavissa: [http://www.nytimes.com/2011/04/23/technology/23cloud.html?\\_r=2](http://www.nytimes.com/2011/04/23/technology/23cloud.html?_r=2)

- LucidChart. 2011. Online Diagram & Flowchart Software - Collaboration Tool | LucidChart. [www-dokumentti]. LucidChart, LLC. [Viitattu 30.3.2011]. Saatavissa: <http://www.lucidchart.com/>
- Masalin, T. 2010. Apple häätää Flashin Maceista. [www-dokumentti]. Tietokone. [Viitattu 19.3.2011]. Saatavissa: [http://www.tietokone.fi/uutiset/apple\\_haataa\\_flashin\\_maceista](http://www.tietokone.fi/uutiset/apple_haataa_flashin_maceista)
- Paananen, P. 2008. Flash-julkaisijan opas. Jyväskylä: WSOY.
- PCWorld. 2008. Web 4.0 Era Is Upon Us. [www-dokumentti]. PCWorld. [Viitattu: 28.3.2011]. Saatavissa: [http://www.pcworld.com/article/143110/web\\_40\\_era\\_is\\_upon\\_us.html](http://www.pcworld.com/article/143110/web_40_era_is_upon_us.html)
- Peltomäki, J. 2001. JavaScript. Jyväskylä: Docendo Finland Oy.
- PHPFrameworks.com. 2007. Top 10 Ranking PHP Frameworks. [www-dokumentti]. PHPFrameworks.com. [Viitattu 30.3.2011]. Saatavissa: <http://www.phpframeworks.com/top-10-php-frameworks/>
- Poikonen, M. M. 2007. Rikkaat Internet-sovellukset. [pdf-dokumentti]. Jyväskylän yliopisto. Kandidaatintutkielma. [Viitattu 27.3.2011]. Saatavissa: [http://users.jyu.fi/~jorma/kandi/2007/Kandi\\_Poikonen.pdf](http://users.jyu.fi/~jorma/kandi/2007/Kandi_Poikonen.pdf)
- Rantala, A. 2005. Web-ohjelmointi. Jyväskylä: Docendo Finland Oy.
- Reyes, J. 2011. Discussing PHP Frameworks: What, When, Why and Which? [www-dokumentti]. Smashing Media GmbH. [Viitattu 30.3.2011]. Saatavissa: <http://www.noupe.com/php/discussing-php-frameworks.html>
- Räihä, K. 2010. Flash on nyt Chromen vakio-ominaisuus. [www-dokumentti]. AfterDawn. [Viitattu 18.3.2011]. Saatavissa: [http://fin.afterdawn.com/uutiset/artikkeli.cfm/2010/06/26/flash\\_on\\_nyt\\_chromen\\_vakio\\_ominaisuus](http://fin.afterdawn.com/uutiset/artikkeli.cfm/2010/06/26/flash_on_nyt_chromen_vakio_ominaisuus)
- Salo, I. 2010. Cloud computing – palvelut verkossa. Porvoo: WSOY-pro Oy
- Sharma, M. 2011. Programming languages on the internet. [www-dokumentti]. Web Developers Notes. [Viitattu 3.3.2011]. Saatavissa: [http://www.webdevelopersnotes.com/basics/languages\\_on\\_the\\_internet.php3](http://www.webdevelopersnotes.com/basics/languages_on_the_internet.php3)

- Siikaniemi, T. 2008. Palvelukeskeisen arkkitehtuurin käyttöönoton lähestymistapoja. [pdf-dokumentti]. Jyväskylän yliopisto. Kandidaatintutkielma. [Viitattu 29.5.2011]. Saatavissa: [users.jyu.fi/~jorma/kandi/2008/Kandi\\_Siikaniemil.pdf](http://users.jyu.fi/~jorma/kandi/2008/Kandi_Siikaniemil.pdf)
- Softpedia. 2011. Joli OS 1.2. [www-dokumentti]. Softpedia. [Viitattu 29.3.2011]. Saatavilla: <http://linux.softpedia.com/get/System/Operating-Systems/Linux-Distributions/Joli-OS-48826.shtml>
- Solis, B. 2008. Introducing The Conversation Prism. [www-dokumentti]. Brian Solis Defining the convergence if media and influence. [Viitattu 18.4.2011]. Saatavissa: <http://www.briansolis.com/2008/08/introducing-conversation-prism/>
- Spivack, N. 2007. How the WebOS Evolves? [www-dokumentti]. Nova Spivack – Minding the Planet. [Viitattu 28.3.2011]. Saatavissa: <http://www.novaspivack.com/technology/how-the-webos-evolves>
- Talouselämä. 2006. Hyvästi, Web 2.0. Tervetuloa, Web 3.0. [www-dokumentti]. Talouselämä. [Viitattu 19.2.2011]. Saatavissa: <http://www.talouselama.fi/uutiset/article162309.ece>
- Tilannehuone.fi. 2011. Tilannehuone.fi – Hälytyskartta. [www-dokumentti]. Tilannehuone.fi. [Viitattu 5.5.2011]. Saatavissa: <http://www.tilannehuone.fi/halytysmap.php>
- Tirronen, M. 2008. Web 2.0: Verkon numerologia. Helsinki: BTJ Kustannus.
- Vertaa.fi. 2011. Microsoft Visio Standard 2010 hinnat – ohjelmistot VERTAA.fi –sivustolla. [www-dokumentti]. Vertaa.fi. [Viitattu 30.3.2011]. Saatavissa: [http://www.vertaa.fi/ohjelmistot/microsoft/visio\\_standard\\_2010/](http://www.vertaa.fi/ohjelmistot/microsoft/visio_standard_2010/)
- W3C. 2009. W3C Mission. [www-dokumentti]. W3C. [Viitattu 28.3.2011]. Saatavissa: <http://www.w3.org/Consortium/mission.html>
- W3C. 2011a. HTML5. [www-dokumentti]. W3C. [Viitattu 28.3.2011]. Saatavissa: <http://www.w3.org/TR/html5/>
- W3C. 2011b. W3C HTML5 Logo. [www-dokumentti]. W3C. [Viitattu: 29.3.2011]. Saatavissa: <http://www.w3.org/html/logo/>
- W3Techs - World Wide Web Technology Surveys. 2011. Usage of JavaScript libraries for websites [www-dokumentti]. Q-Success. [Viitattu 30.3.2011]. Saatavissa: [http://w3techs.com/technologies/overview/javascript\\_library/all](http://w3techs.com/technologies/overview/javascript_library/all)

Weaver, E. 2006. Web 2.0 names that hurt. [www-dokumentti]. Brand Dialogue LLC. [Viitattu 21.4.2011]. Saatavissa: <http://www.branddialogue.com/diabloque/2006/06/names-that-hurt/>

Zandstra, M. 2001. PHP. Helsinki: Oy Edita Ab.