

Opinnäytetyö (AMK)

Tietojenkäsittely

Tietojärjestelmät

2010

Petri Airinen

PILVILASKENTA JA PILVIPALVELUT

– Pilvialustojen vertailu



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittelyn koulutusohjelma | Tietojärjestelmät

Maaliskuu 2011 | Sivumäärä 51

Ohjaaja Anne Jumppanen

Petri Airinen

PILVILASKENTA JA PILVIPALVELUT

Pilvilaskennasta puhutaan paljon ja usein kuulee, että ollaan siirtymässä pilvilaskennan tai pilvipalveluiden käyttäjäksi. Pilvilaskennassa resurssit, kapasiteetti tai ohjelmistot siirretään käytettäväksi internetin kautta. Pilvilaskentaa käyttäessä ei tarvitse hankkia omia laitteita, vaan ne voidaan vuokrata internetin välityksellä pilvestä. Pilvellä tarkoitetaan käytännössä internetiä. Pilvilaskennan houkuttelevimpia ominaisuuksia on sen skaalautuvuus, nopea käyttöönotto, hinnoittelutavat ja lähes rajaton kapasiteetti.

Työssä tutkitaan kolmea eri pilvialustaa, jotka ovat Windows Azure, Amazon Web Services ja Google App Engine. Työn tavoitteena on selvittää tutkittavana olevien pilvialustojen ominaisuuksia ja mihin tarkoitukseen ne sopivat. Työssä selvitetään myös pilvilaskentaan liittyviä käsitteitä kuten IaaS (Infrastructure as a Service), PaaS (Platform as a Service) ja SaaS (Software as a Service).

Vertailuosuudessa tarkastellaan miten nämä eri pilvialustat eroavat toisistaan ominaisuuksiltaan ja mitä rajoitteita niillä on. Vertailussa tarkastellaan myös pilvialustojen hinnoittelua.

Työn tuloksissa ja johtopäätöksissä selvitetään mihin tutkittavana olevat pilvialustat soveltuvat ja mitä niillä voidaan tehdä. Tulokset voivat helpottaa päätöksen tekoa siirryttäessä pilvilaskennan tai pilvipalveluiden käyttäjäksi.

ASIASANAT:

Pilvilaskenta, IaaS, PaaS, SaaS

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | Information Systems

March 2011 | Total number of pages 51

Instructor Anne Jumppanen

Petri Airinen

CLOUD COMPUTING AND CLOUD SERVICES

Cloud computing is becoming more popular each day and many companies are starting to use cloud computing or cloud services. In cloud computing, resources, capacity or software are transferred to the cloud and used via Internet. Cloud computing removes the need to acquire own hardware. Instead the hardware is rented via Internet from the cloud. The term cloud basically means the Internet. The key features of cloud computing include its scalability, fast deployment, pricing models and nearly unlimited capacity.

This thesis focuses on three different cloud computing platforms i.e. Windows Azure, Amazon Web Services and Google App Engine. The goal of this thesis was to research these cloud computing platforms and find out their features and also what purpose they are best suited for. The thesis also explains the terms IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service).

In the comparison part of this thesis, the properties and limitations of the cloud platforms are compared with each other. The pricing policies of cloud platforms are also compared.

The results and conclusions answer to the questions, what the cloud computing platforms are suitable for and what can you do with them. The results may ease the decision whether to use cloud computing or cloud services.

KEYWORDS:

Cloud computing, IaaS, PaaS, SaaS

SISÄLTÖ

1 JOHDANTO	5
2 PILVILASKENTA	7
2.1 Määritelmä	7
2.2 Pilvilaskenta yleisesti	7
2.3 Keskeisiä käsitteitä	8
2.4 Pilvilaskennan jaottelu	9
2.4.1 IaaS	9
2.4.2 PaaS	11
2.4.3 SaaS	13
2.4.4 Vertailua	15
3 WINDOWS AZURE PLATFORM	17
3.1 Windows Azure	17
3.2 Windows Azure Storage	18
3.3 Sisällönjakeluverkko	18
3.4 SQL Azure	19
3.4.1 Hallittavuus	20
3.4.2 Skaalautuvuus	20
3.4.3 Ohjelmistokehitys	21
3.5 Windows Azure AppFabric	21
3.5.1 Palveluväylä	21
3.5.2 Pääsynhallinta	22
3.5.3 Välimuistipalvelu	23
4 AMAZON WEB SERVICES	24
4.1 Amazon EC2	24
4.1.1 Elastic Block Store	25
4.1.2 Amazon CloudWatch	25
4.1.3 Auto Scaling	26
4.1.4 Elastic Load Balancing	26
4.1.5 Korkean suorituskyvyn laskenta	27
4.2 Amazon S3	27
4.2.1 Ominaisuuksia	28
4.2.2 Tiedon suojaus	29
5 GOOGLE APP ENGINE	30
5.1 Sovellusympäristö	31

5.1.1 Hiekkalaatikko	31
5.1.2 Java-ympäristö	32
5.1.3 Python-ympäristö	32
5.1.4 Datastore	32
5.1.5 Google-tilit	33
5.1.6 App Engine -palvelut	33
5.2 Ohjelmistokehitys	35
5.3 Kiintiöt ja rajoitukset	36
6 PILVITEKNOLOGIA-ALUSTOJEN VERTAILU	38
6.1 Hinnoittelu	38
6.1.1 Windows Azure Platform	38
6.1.2 Amazon Web Services	41
6.1.3 Google App Engine	44
6.2 Ominaisuudet	45
7 TULOKSET JA JOHTOPÄÄTÖKSET	47
7.1 Windows Azure Platform	47
7.2 Amazon Web Services	48
7.3 Google App Engine	49
LÄHTEET	50
KUVIOT	
Kuvio 1. IaaS-palvelumalli.....	10
Kuvio 2. PaaS-palvelumalli	12
Kuvio 3. SaaS-palvelumalli	14
TAULUKOT	
Taulukko 1. Windows Azuren hinnoittelu	39
Taulukko 2. SQL Azuren hinnoittelu.....	39
Taulukko 3. Windows Azure AppFabric hinnoittelu	40
Taulukko 4. Windows Azuren tiedonsiirron hinnoittelu	40
Taulukko 5. Amazon EC2 -instanssien hinnoittelu	41
Taulukko 6. Amazon internet-tiedonsiirron hinnoittelu	42
Taulukko 7. Amazon Elastic Block Storen hinnoittelu.....	42
Taulukko 8. Amazon Elastic Load Balancing hinnoittelu	43
Taulukko 9. Amazon S3 hinnoittelu.....	44
Taulukko 10. Google App Enginen hinnoittelu	45
Taulukko 11. Pilvialustojen ominaisuuksia	45

1 Johdanto

Pilvilaskennasta puhutaan tällä hetkellä paljon ja yhä useammin kuulee, että ollaan siirtymässä pilveen. Pilvilaskennassa resurssit, kapasiteetti tai ohjelmistot siirretään käytettäväksi internetin kautta, eli toisin sanoen laitteisto, laskentateho ja käytetyt ohjelmat saadaan internetistä. Pilvilaskentaa käyttäessä yrityksen ei tarvitse hankkia esimerkiksi omia konesaleja tai laitteita, vaan ne voidaan vuokrata internetin välityksellä pilvestä. Pilvellä tarkoitetaan käytännössä internetiä.

Pilvilaskennan ehkä houkuttelevimpia ominaisuuksia ovat sen skaalautuvuus, nopea käyttöönotto, hinnoittelutavat ja lähes rajaton kapasiteetti.

Opinnäytetyössä käsitellään kolmea eri pilvialustaa (Cloud Platform), jotka ovat Windows Azure, Amazon Web Services ja Google App Engine. Työn tavoitteena on selvittää näiden kolmen pilvialustan ominaisuuksia ja tutkia, minkälaiseen tarkoitukseen ne soveltuvat parhaiten. Työssä selvitetään myös pilvilaskentaan liittyviä käsitteitä kuten IaaS (Infrastructure as a Service), PaaS (Platform as a Service) ja SaaS (Software as a Service).

Vertailuosuudessa tarkastellaan miten nämä eri alustat eroavat toisistaan ominaisuuksiltaan ja mitä mahdollisia rajoitteita niillä on. Vertailussa tarkastellaan myös eri alustojen hinnoittelua.

Opinnäytetyö on tehty toimeksiantona Luu5 Oy:lle. Luu5 Oy on kolme henkeä työllistävä ohjelmistoalan yritys Turusta, joka on erikoistunut SaaS-ohjelmistojen tietokanta- ja sovellusarkkitehtuuriin, ansaintalogiikkaan ja tuotekehitykseen.

Työssä tarkastellaan ensin mitä pilvilaskenta on ja mitä siihen sisältyy. Tämän jälkeen tutkitaan tarkasteltavana olevat pilvialustat. Pilvialustojen tutkimisen jälkeen niitä vertaillaan toisiinsa niin ominaisuuksiltaan kuin hinnaltaan. Lopuksi tarkastellaan yhteenveto-osiossa tärkeimpiä asioita, joita ilmaantui pilvialustoja tutkittaessa ja mihin tarkoitukseen tutkittavana olevat kolme pilvialustaa sopivat

parhaiten. Työn tuloksien pohjalta toimeksiantajan on helpompi tehdä päätös, mitä pilvialustaa tullaan mahdollisesti itse käyttämään tulevaisuudessa.

2 Pilvilaskenta

2.1 Määritelmä

Pilvilaskennasta käytetään yleensä termejä pilvilaskenta (cloud computing) tai pilvipalvelut (cloud services). Nämä kuitenkin tarkoittavat hieman eri asioita.

IDC Exchange -sivustolla Frank Gensin kirjoittamassa artikkelissa nämä kaksi termiä on määritelty seuraavalla tavalla:

“Cloud Services = Consumer and Business products, services and solutions that are delivered and consumed in real-time over the Internet” (Gens 2008).

“Cloud Computing = an emerging IT development, deployment and delivery model, *enabling* real-time delivery of products, services and solutions over the Internet (i.e., enabling cloud services)” (Gens 2008).

Pilvipalvelut ovat siis kuluttaja- tai yritysmaailman tuotteita, palveluita tai ratkaisuja, joita tarjotaan ja käytetään internetin välityksellä reaaliajassa (SearchCloudComputing 2007).

Pilvilaskenta sen sijaan keskittyy enemmän tekniselle puolelle. Pilvilaskenta sisältää tekniikat ja teknologiat, joiden avulla esimerkiksi ohjelmistokehittäjät voivat julkaista tuotteensa tai palvelunsa internetissä (SearchCloudComputing 2007).

2.2 Pilvilaskenta yleisesti

Pilvi-sanaa käytetään vertauskuvana internetistä, koska tietokoneverkkokaavioissa internet kuvataan pilvenä, joka on abstrakti käsite monimutkaisesta infrastruktuurista, jonka se peittää. Pilvilaskennassa toiminnallisuudet tarjotaan palveluna internetin välityksellä. Pilvilaskennan käyttöönottoaminen on helppoa, koska siihen tarvittava infrastruktuuri on jo valmis ja toimintakunnossa. On olemassa olevia tietokeskuksia, joista voi vuokrata laskentatehoa ja tallennustilaa. (Azure Academic Pilot 2010a.) Tällaisia tietokeskuksia on muun muassa Microsoftilla, Amazonilla ja Googella. Kyseisillä yrityksillä on tietokeskuksia useilla mantereilla ja alueilla, minkä

vuoksi voidaan palvella useassa maassa olevaa asiakasta ilman, että latenssi kasvaisi suureksi kaukaisimmissa paikoissa.

Päämotivaattoreina pilvilaskennan kasvussa on sen hinnoittelu ja ketteryys. Pilvilaskennalla voidaan varautua ruuhka-aikoihin, jolloin kapasiteetin tarve on huomattavasti suurempi kuin normaalisti. Pilvilaskennalla pyritään antamaan kapasiteettia vain sen verran, kuin sitä tarvitaan. Tämän ansiosta vältetään käyttämättömältä kapasiteetilta, joka on yksi ongelmista yrityksissä, joissa ei pilvilaskentaa käytetä. (Verizon 2009.)

Pilvilaskennassa kapasiteetti on siis skaalautuva eli kapasiteettia saadaan lisää tarpeen mukaan. Tuloksena saadaan ketteryyttä ja pienemmät kustannukset. (Azure Academic Pilot 2010a.)

Etenkin pienille ja aloittaville yrityksille pilvilaskennan hyödyntäminen on vartenotettava vaihtoehto. Kun esimerkiksi omaa konesalia ei tarvitse hankkia, tehdään suuria säästöjä alkuinvestoinneissa. Pilvilaskennan käyttöönotto on nopeaa, joten yritysten pääsy ideasta tuotteeseen nopeutuu.

Pilvilaskennan käyttö tuo turvaa myös silloin, kun yrityksen tekemä tuote tai palvelu ei menesty. Pienten alkuinvestointien ja nopean markkinoille pääsyn ansiosta menetykset pysyvät kohtuullisen vähäisinä.

2.3 Keskeisiä käsitteitä

Palvelutasosopimus

Palvelutasosopimus eli SLA (Service Level Agreement) on asiakkaan ja palveluntarjoajan välinen sopimus, jossa usein palveluntarjoaja takaa tietyn saatavuustason palvelulleen. Jos sovitun sopimuksen rajoissa ei pysytä, seuraa siitä usein jokin yhteisesti sovittu sanktio. (Luu5 2010.)

Skaalautuvuus

Skaalautuvuudella tarkoitetaan tässä yhteydessä esimerkiksi järjestelmän tai sovelluksen kykyä kasvaa kysynnän kasvaessa. Pilvisovelluksissa skaalautuvuus on yksi keskeisimmistä käsitteistä. (WebDAM 2011.)

Saatavuus

Saatavuudella tarkoitetaan sitä, kuinka varmasti järjestelmä, laite, ohjelma tai palvelu on tarvittaessa käytettävissä. Saatavuutta heikentäviä tekijöitä voi olla esimerkiksi tietojärjestelmän virheet, joiden vuoksi järjestelmää ei voida käyttää. Saatavuutta voi haitata myös verkkoyhteyksien häiriöt. (Sanastokeskus TSK ry 2002.)

Monikäyttäjä

Monikäyttäjät (multi-tenant) on pilvilaskennassa käytetty termi, jolla tarkoitetaan käytännössä sitä, että esimerkiksi jokin pilvessä oleva sovellus palvelee useaa käyttäjää samanaikaisesti. Monikäyttäjäsovelluksessa samaa sovellusta voi siis käyttää monta käyttäjää samanaikaisesti. (MSDN 2006.)

2.4 Pilvilaskennan jaottelu

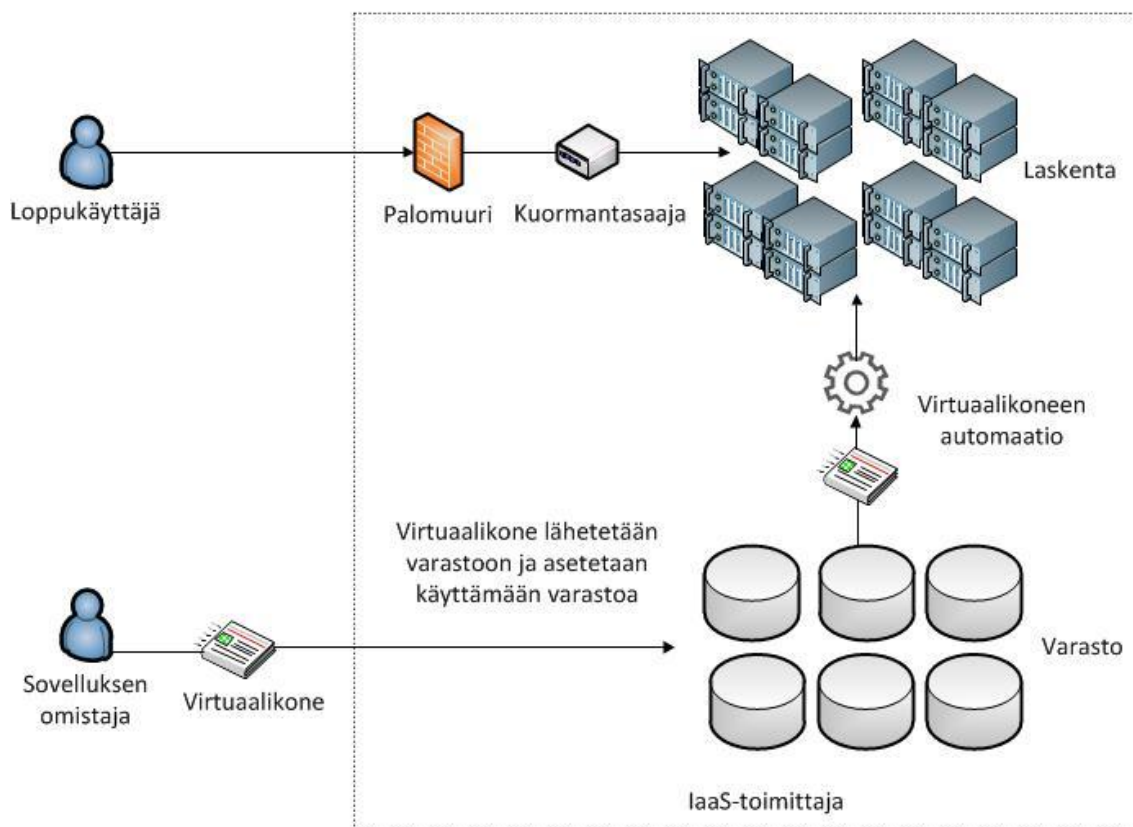
Pilvilaskenta on jaoteltu kolmeen eri osaan, joita ovat IaaS, PaaS ja SaaS. Nämä kolme mallia on tarkoitettu käytettäväksi erilaisten palvelujen ja ratkaisujen aikaansaamiseksi.

2.4.1 IaaS

IaaS (Infrastructure as a Service) -palvelumallilla tarjotaan internetin välityksellä laskentatehoa, säilytystilaa ja verkkoinfrastruktuuria, kuten esimerkiksi palomureja ja kuormantasaajia. Tyypillinen IaaS-palvelua käyttävä asiakas haluaa ajaa sovellustaan palveluntarjoajan palvelimella. (SearchCloudComputing 2009) Alunperin tämän tyyppistä tarjontaa kutsuttiin termillä Hardware as a Service. Nykyään käytetty termi on IaaS. (Pijanowski 2009.)

IaaS-palveluja tarjoavat yritykset käyttävät virtualisointitekniikkaa laskentatehon tarjoamiseen. Näin ollen palvelu siis tarjotaan virtuaalikoneita käyttäen. Palvelun käyttäjä tekee tarvittavat asennukset virtuaalikoneelle, minkä jälkeen sovellusta voidaan käyttää internetin välityksellä virtuaalipalvelimen kautta. Asiakkaan sovellus ja kaikki mitä se tarvitsee toimiakseen, täytyy asentaa

virtuaalikoneelle. Tämän jälkeen virtuaalikone siirretään palvelun tarjoajan palvelinympäristöön, jossa se täytyy asettaa käyttämään palvelun tarjoajan tietovarastoa. Asetusten teon jälkeen virtuaalikone voidaan käynnistää ja sen jälkeen loppukäyttäjä pääsee käsiksi sovellukseen. Sovelluksen tarvittavan levytilan ja laskentatehon tarjoaa palvelun tarjoaja. (Pijanowski 2009.)



Kuvio 1. IaaS-palvelumalli

Kuviossa 1 on kuvattu IaaS-palvelumalli. Sovelluksen omistaja siirtää virtuaalikoneen IaaS-toimittajan palvelimelle ja asettaa sen käyttämään toimittajan palvelimella olevaa varastoa.

Virtuaalikoneen automaatio hoitaa tarvittavat toimenpiteet. Virtuaalikone käyttää IaaS-toimittajan laskentatehoa.

Loppukäyttäjä näkee valmiin sovelluksen ja voi käyttää valmista sovellusta internetin välityksellä. Käyttäjän yhteys menee IaaS-toimittajan palomuurien ja mahdollisten kuormantasaajien läpi, minkä jälkeen sovellus on loppukäyttäjän käytettävissä.

IaaS-palvelumalli on kaikista pilvilaskennan palvelumalleista joustavin. Tämän tyyppisellä mallilla tarjotaan erilaista infrastruktuuria palveluna, kuten esimerkiksi palvelimia ja varastoja (kiintolevytilaa).

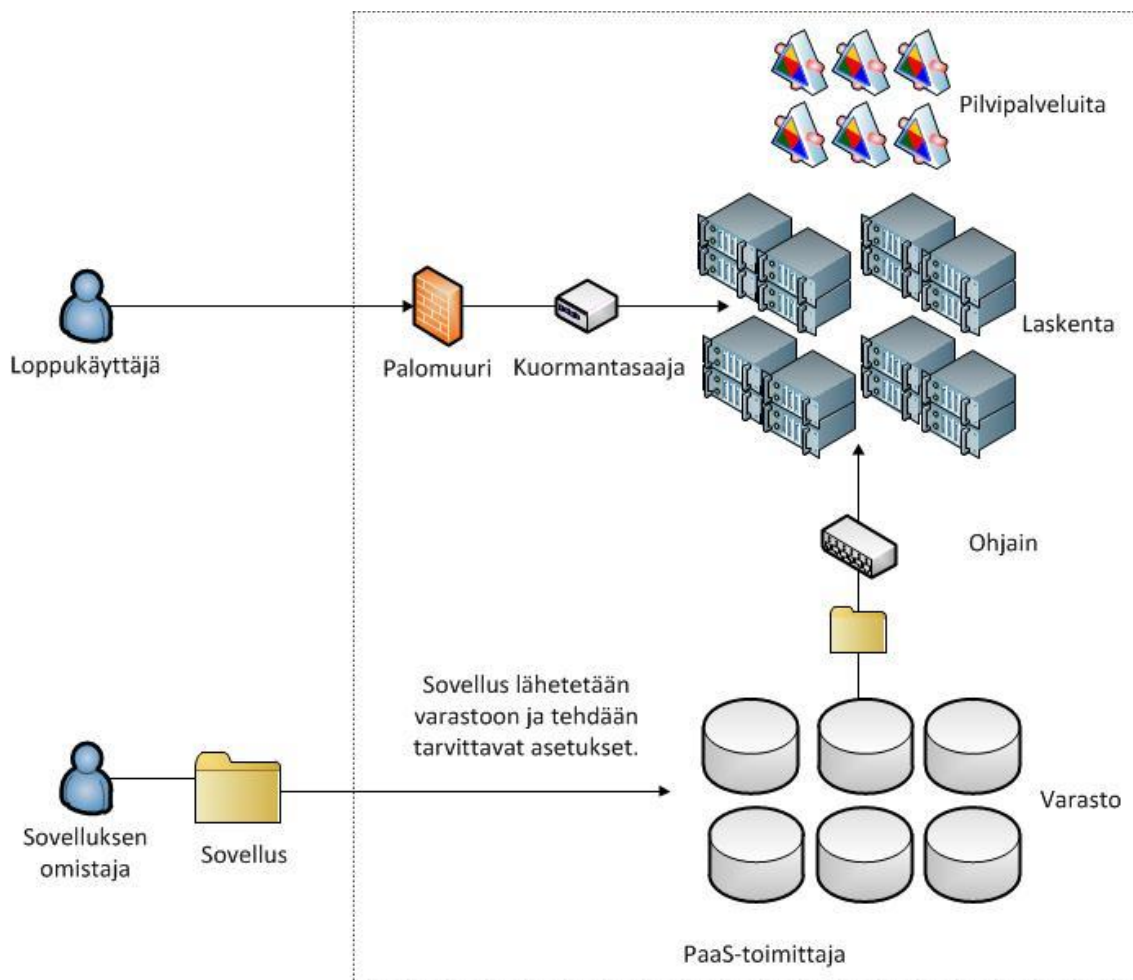
IaaS on hyvä vaihtoehto silloin, kun ei ole aikaa muuttaa sovellusta pilviympäristöön toimivaksi. Sovellus voidaan siirtää IaaS-mallilla pilveen ilman muutoksia sovellukseen. Pilvessä olevaan virtuaalikoneeseen tarvitsee asentaa tarvittavat ohjelmistot ja järjestelmät, että sovellus voidaan ajaa aivan kuin se olisi yrityksen omilla palvelimilla. (Qrimp 2008.)

2.4.2 PaaS

PaaS (Platform as a Service) -palvelumallilla tarjotaan myös laskentatehoa, säilytystilaa ja verkkoinfrastruktuuria. Tämän lisäksi PaaS tarjoaa ajoympäristön ohjelmointikoodille. Tämä tarkoittaa sitä, että koko virtuaalikonetta ei tarvitse rakentaa, asentaa ja lähettää palveluntarjoajalle. Pelkästään ohjelmistokoodi lähetetään ja käynnistetään. PaaS-palvelua käyttävä asiakas on henkilö tai yritys, joka tarvitsee palvelinympäristön sovellukselleen. (Qrimp 2008.)

PaaS-ympäristö tarjoaa laskentatehoa ajoympäristön välityksellä. Näin ollen palvelu tarjotaan pakettina, joka sisältää ohjelmistokoodin tai jonkin käännetyn version ohjelmistokoodista. (Pijanowski 2009.)

Ohjelmistopakettien kooditiedostot ohjelmoidaan PaaS-tarjoajan tarjoamalla ohjelmointiympäristöllä. Esimerkiksi Windows Azure Platform tukee muun muassa .NET frameworkia ja PHP:tä. Toisena esimerkkinä Googlen App Engine, joka tukee Javaa ja Pythonia. Useimmat PaaS-palveluntarjoajat tarjoavat lisäksi myös pilvipalveluita, joiden avulla pystyy muun muassa tallentamaan tietoa, ottamaan yhteyttä muihin palveluihin, laskuttamaan asiakkaita ja hoitamaan käyttäjän tunnistukset. Näitä pilvipalveluita tarvitaan, koska kolmannen osapuolen komponentteja ei pysty asentamaan. (Pijanowski 2009.)



Kuvio 2. PaaS-palvelumalli

Kuviossa 2 on kuvattu PaaS-palvelumalli. Sovelluksen omistaja siirtää sovelluksen PaaS-toimittajan tietovarastoon ja tekee tarvittavat asetukset.

PaaS-toimittajan palvelimella oleva ohjain käynnistää sovelluksen ja välittää käskyjä.

Loppukäyttäjä näkee valmiin sovelluksen ja voi käyttää valmista sovellusta internetin välityksellä. Käyttäjän yhteys menee PaaS-toimittajan palomuurien ja mahdollisten kuormantasaajien läpi, minkä jälkeen sovellus on loppukäyttäjän käytettävissä.

Sovellus voi käyttää myös palveluntarjoajan tarjoamia pilvipalveluita. Tällaisia palveluita voidaan yhdistää omaan sovellukseen. Tällaisia pilvipalveluita ovat esimerkiksi Googlen käyttäjätilit ja Gmail.

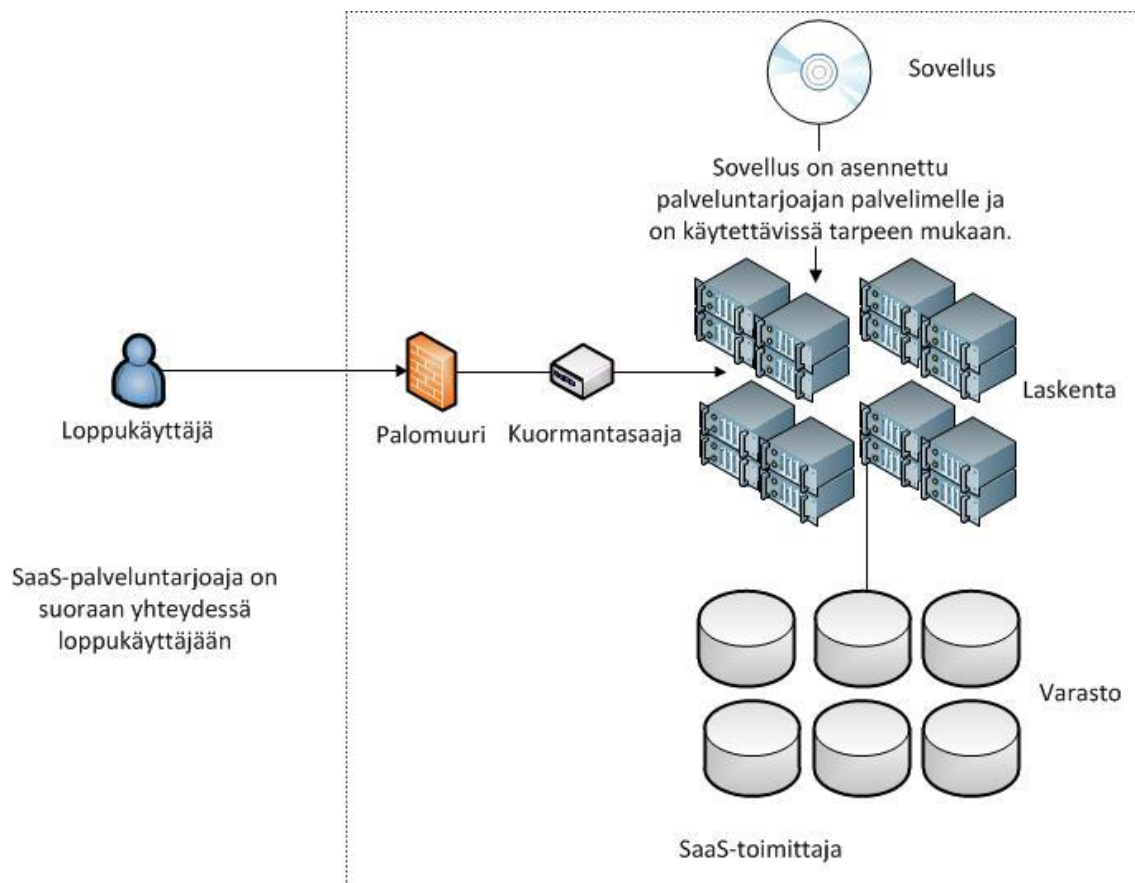
PaaS-ympäristön ominaisuuksiin kuuluu myös se, että skaalautuvuus voidaan määritellä asetusten välityksellä, eikä sitä tarvitse ohjelmoida sovellukseen. Vaikka PaaS-ympäristöillä voidaan päästä eroon kolmannen osapuolen ohjelmistoista ja skaalautuvuuden pystyy hoitamaan helpommin, sillä on myös haittapuolensa. Ohjelmointi on tehtävä ympäristön vaatimalla tavalla ja se saattaa lukita käyttämään tiettyä PaaS-palveluntarjoajaa. (Pijanowski 2009.)

Ajonaikaisen ympäristön ja pilvipalveluiden lisäksi PaaS-palveluntarjoaja voi tarjota interaktiivisen kehitysympäristön, minkä ansiosta kehittäjä voi ohjelmoida myös offline-tilassa. Interaktiivinen kehitysympäristö voi myös sisältää työkaluja, joiden vuoksi voidaan paketoita ohjelma ja lähettää se suoraan PaaS-ympäristöön kehitystyökalun kautta. (Ciol 2010.)

2.4.3 SaaS

SaaS (Software as a Service) on tapa toimittaa sovelluksia internetin välityksellä palveluna. Toisin kuin perinteisen tavan mukaan, jossa ohjelma itse asennetaan ja hallinnoidaan, SaaS-mallilla toteutettua sovellusta päästään käyttämään internetselaimen kanssa ilman, että joudutaan itse asentamaan ohjelmaa. SaaS-sovellus toimii toimittajan palvelimilla. Sovelluksen toimittaja huolehtii palvelun turvallisuudesta, saatavuudesta ja toimivuudesta. (Bhaskardeep 2010.)

Käyttäjien ei tarvitse huolehtia laitteiston tai ohjelmiston hankkimisesta, asennuksista, ylläpidosta tai päivityksistä. Käyttäjä tarvitsee vain internet-yhteyden, jotta sovellukseen pääsee käsiksi. Tämän tyyppistä sovellusta voivat käyttää samanaikaisesti tuhannet käyttäjät. Käyttäjän ei tarvitse investoida etukäteen palvelimiin tai lisensseihin. Toimittajalla on vain yksi sovellus hallinnoitavana, mikä vähentää kustannuksia. (Bhaskardeep 2010.)



Kuvio 3. SaaS-palvelumalli

Kuviossa 3 on kuvattu SaaS-palvelumallin toimintaa. SaaS-palveluntarjoaja tarjoaa sovellusta loppukäyttäjälle. Loppukäyttäjä ottaa yhteyden sovellukseen internetin välityksellä. Yhteys menee palveluntarjoajan palomuurien ja kuormantasaajien läpi.

Sovellus toimii palveluntarjoajan palvelimilla. Palveluntarjoaja tarjoaa tarvittavan laskentatehon ja varastotilan, jotta ohjelma voi toimia.

SaaS-toimittaja omistaa ohjelman, joka on käynnissä ja saatavilla tarpeen mukaan. SaaS-toimittaja on myös vastuussa palvelinympäristöstä. (Pijanowski 2009.)

SaaS:n arvo tulee siitä, että kokonaisia ja valmiita sovelluksia saa käyttöönsä internetin välityksellä. Loppukäyttäjän ei tarvitse hankkia itselleen lisenssejä tai ylläpitää sovellusta vanhan tavan mukaan, jossa sovelluksiin yleensä hankittiin lisenssit ja esimerkiksi ohjelmistopäivitykset oli itse asennettava.

Loppukäyttäjän ei tarvitse myöskään ostaa ja ylläpitää erillistä infrastruktuuria, jonka päällä sovellusta ajetaan. Toisin sanoen palvelun tarjoaja huolehtii laskentatehosta ja varastotilasta. Sovelluksen ajamiseen käyttäjä tarvitsee kuitenkin tietokoneen tai mobiililaitteen, joka on tarpeeksi tehokas ajamaan kyseistä sovellusta. Hinnoittelu SaaS-tyyppisissä sovelluksissa on yleensä yhden käyttäjän hinta kuukausittain. (Pijanowski 2009.)

2.4.4 Vertailua

Kaikkein joustavin näistä malleista on IaaS, jolla pystyy käytännössä tekemään kaiken saman mitä kahdella muulla mallilla voi tehdä. Toisena etuna voidaan nähdä se, että se ei "lukitse" mihinkään tiettyyn alustaan tai ohjelmointikieleen, koska siinä vuokrataan virtuaalikoneita, joille itse asennetaan tarvittavat ohjelmat ja työkalut. Haittapuoleksi voidaan katsoa se, että se on muita malleja työläämpi. (Learning Tree 2010.)

PaaS-palvelumalli soveltuu parhaiten kehittäjille, koska siinä tarjotaan kehitysympäristöä. Tällä palvelumallilla ei tarvitse itse huolehtia ohjelmistojen asennuksista vaan riittää, että ohjelmoidaan sovellus ja lähetetään se PaaS-toimittajan palvelimelle. (Zoho Creator 2011.) Tämän palvelumallin haittapuoleksi voidaan katsoa se, että se voi "lukita" yrityksen tai asiakkaan käyttämään samaa palveluntarjoajaa. PaaS-toimittajien alustoilla joudutaan ohjelmoimaan palveluntarjoajan ehtojen mukaan, eli toisin sanoen niillä ohjelmointikielillä, joita palveluntarjoajan kehitysalusta tukee.

SaaS-palvelumallilla yritys tai sovelluksen omistaja tarjoaa sovellustaan käytettäväksi internetin välityksellä (Bhaskardeep 2010). IaaS- ja PaaS-palvelumalleilla voidaan toteuttaa SaaS-sovelluksia. SaaS-sovelluksien toteuttaminen ei kuitenkaan vaadi IaaS- tai PaaS-palvelumallien käyttämistä.

Usein IaaS- ja PaaS-toimittajat tarjoavat myös omia pilvipalveluitaan, joita asiakkaat voivat käyttää hyväksi omissa ratkaisuisaan. Nämä palvelut voivat olla joko maksullisia tai ilmaisia. Esimerkiksi Google tarjoaa muun muassa Gmailia ja käyttäjätilejä, ja niitä voidaan käyttää esimerkiksi käyttäjien

tunnistamiseen omissa ratkaisuissa. Amazon tarjoaa myös ison valikoiman palveluita, joista yksi hyvä esimerkki on S3 (Amazon Simple Storage Service) varastopalvelu, jonka voi yhdistää muihin Amazonin palveluihin. Windows Azure tarjoaa muun muassa integraatiota Windows Live -palveluihin ja SQL Azurea.

3 Windows Azure Platform

Windows Azure Platform on PaaS (Platform as a Service) -tyyppinen alusta, joka tarjoaa internetin välityksellä käytettävän käyttöjärjestelmän ja kehityspalveluita, joita voi käyttää itsenään tai yhdessä muiden alustan työkalujen kanssa. Windows Azure Platformille voidaan ohjelmoida sovelluksia tai sillä voidaan tehostaa olemassa olevia sovelluksia, jotka toimivat pilvessä tai jolla on pilvelle suunnattuja toiminnallisuuksia. Windows Azurella voidaan ohjelmoida sovelluksia tietokoneisiin, mobiililaitteisiin ja palvelimiin. Sillä voidaan ohjelmoida myös web ja hybridisovelluksia. Hybridisovellus voi olla esimerkiksi sellainen sovellus, joka on tietokoneelle asennettava, mutta se käyttää joitakin pilven palveluita. (Azure Academic Pilot 2010b.)

3.1 Windows Azure

Windows Azure on pilvipalvelu-käyttöjärjestelmä, joka toimii kehitys-, palvelun tarjoamis- ja hallintaympäristönä Windows Azure Platformille (Azure Academic Pilot 2010b). Esimerkiksi fyysiset resurssit (laitteisto) on viety pilveen, josta pilvisovellukset voivat käyttää niitä tarpeen mukaan. Säilytystila on myös viety pilveen. Fyysinen laitteisto ja sovellusalusta on abstraktoitu ja se tarjotaan pilvessä virtualisoituna laskentatehona ja levytilaresursseina. Lisäksi jokaista sovelluksen instanssin saatavuutta ja skaalautuvuutta tarkastellaan ja hallinnoidaan automaattisesti. Windows Azure toimii Microsoftin tietokeskuksissa. (Azure Academic Pilot 2010c.) Windows Azure tarjoaa kehittäjälle tarpeen mukaan laskentatehoa, levytilaa, skaalautuvuutta ja pilvisovellusten hallintaa (Azure Academic Pilot 2010b).

Windows Azurella voidaan ohjelmoida käyttäen Visual Basic tai C#-ohjelmointikieliä. Windows Azure on avoin alusta, joka tukee myös muita ohjelmointikieliä ja ympäristöjä. Yhteensopivia ohjelmointikieliä ja ympäristöjä ovat muun muassa Eclipse, Ruby, PHP ja Python. (Azure Academic Pilot 2010c.)

3.2 Windows Azure Storage

Windows Azure -varastopalvelut tarjoavat pysyvän ja kestäväen varastotilan pilvessä. Varastopalveluihin sisältyvät seuraavat ominaisuudet:

- Binary Large Object (BLOB) -palvelu tekstin tai binääridatan tallennukseen.
- Jonopalvelu, jonka avulla palvelut voivat viestiä toistensa kanssa.
- Taulukkopalvelu rakenteelliselle varastolle, johon voidaan suorittaa kyselyitä.
- Windows Azure asema, jonka avulla Windows Azure sovellukset voivat ottaa käyttöön Page Blob:n, joka on käytännössä virtuaalinen kiintolevy. Sen avulla sovellukset pystyvät lataamaan tai lähettämään virtuaalilevyjä blobeina. (Windows Azure 2010.)

Windows Azure SDK (Software Development Kit) tarjoaa API:t (Application Programming Interface), joiden avulla voidaan käsitellä varastopalveluita. Varastopalveluihin voi päästä käsiksi Windows Azuressä olevan palvelun kautta, tai suoraan internetin välityksellä sovelluksista, jotka voivat lähettää ja vastaanottaa tietoa HTTP:n ja HTTPS:än välityksellä. (Windows Azure 2010.)

3.3 Sisällönjakeluverkko

Sisällönjakeluverkko eli CDN (Content Delivery Network), tarjoaa integraation Windows Azure -varastoon. Windows Azure CDN tehostaa loppukäyttäjän suorituskykyä ja luotettavuutta asettamalla kopioita tiedosta lähemmäksi käyttäjiä. (Windows Azure 2010b.) Toisin sanoen Windows Azure CDN strategisesti asettaa tietoa lähemmäksi käyttäjiä, jolloin verkkoliikennettä voidaan käyttää tehokkaammin.

Monet Microsoftin tuotteet käyttävät Windows Azure CDN:ää sisällön toimittamiseen. Näitä tuotteita ovat esimerkiksi Windows Update, Zune videot ja Bing kartat. Windows Azure CDN voi toimittaa sisältöä Windows Azure varastossa sijaitsevista julkisista blob-säiliöistä. (Windows Azure 2010b.)

Tuettua sisältöä ovat esimerkiksi

- JPG, CSS ja JavaScript
- mediatiedostot, ohjelmistot ja dokumentit
- sovellukset ja reaaliaikaiset medialähetkset. (Windows Azure 2010b.)

3.4 SQL Azure

SQL Azure tarjoaa Microsoft SQL Serverin toiminnallisuuksia pilvessä web-pohjaisena relaatiotietokantana. Se tarjoaa web-palveluita, jotka mahdollistavat relaationaaliset kyselyt, hakutoiminnot ja tiedon synkronisaation mobiilikäyttäjien, etätoimistojen ja liikekumppaneiden kanssa. (Azure Academic Pilot 2010b.) SQL Azuren kolme keskeistä periaatetta ovat hallittavuus, skaalautuvuus ja ohjelmistokehitys (Lee ym. 2009, 4).

Kehittäjän näkökulmasta SQL Azure tarjoaa tunnetun ja tutun relaatiotietokantamallin, joka muistuttaa hyvin paljon Microsoft SQL Serveriä. SQL Azure yksinkertaistaa sovelluksien luomista, toimittamista ja prototyyppien tekemistä. SQL Azure poistaa infrastruktuuriesteet antaen näin kehittäjälle enemmän vapautta kokeilla uusia tapoja jakaa tietoa. (Lee ym. 2009, 4.) Toisin sanoen ei tarvitse hankkia laitteistoja ja ohjelmistoja tietokantojen pystyttämiseen ja ylläpitämiseen, vaan Microsoft hoitaa tämän puolen.

IT-hallinnon näkökulmasta SQL Azure tarjoaa systemaattisen ja turvallisen pilveen perustuvan ratkaisun, joka integroituu talon sisällä sijaitseviin laitteisiin ja ohjelmistoihin. SQL Azure on rakennettu samojen Microsoft SQL Server -teknologioiden päälle, joita on jo käytetty ja jotka on todistettu toimiviksi talon sisäisissä ratkaisuissa ja joilla on tuotettu korkean saatavuuden, luotettavuuden ja turvallisuuden ratkaisuja. (Lee ym. 2009, 4.)

Liiketoiminnan kannalta SQL Azure tarjoaa tehokkaan lähestymistavan tiedon hallintaan joustavalla kulutukseen pohjautuvalla hinnoittelulla. Se tarjoaa myös kilpailukykyiset aloituskulut ja kyvyn skaalata kapasiteettia tarpeiden mukaan. (Lee ym. 2009, 4.)

3.4.1 Hallittavuus

SQL Azuren tietokanta tarjoaa samankaltaisen saatavuuden ja toiminnallisuudet kuin yrityksen oma tietokeskus, mutta ilman samanlaista hallinnollista puolta, joka tulee esiin yrityksen omissa fyysisissä ratkaisuisissa. (Lee ym. 2009, 5.)

Perinteisellä paikan päällä sijaitsevalla infrastruktuurilla aika, joka menee tietokantojen pystyttämisessä, voi hidastaa prototyyppien testailua ja uusien ratkaisujen markkinoille tuomista. Käyttämällä pilveen pohjautuvia ratkaisuja, kuten SQL Azure, voidaan tietovarastot pystyttää minuuteissa ja mukautua nopeasti muutoksiin. Hankintakustannuksissa tehdään säästöjä, koska maksetaan vain käytettävästä kapasiteetista. Tietokannan kokoa voidaan suurentaa helposti tarvittaessa. (Lee ym. 2009, 5.)

SQL Azure varmuuskopioi tiedot ja pitää niistä useaa kopiota useilla eri palvelimilla, jotta tiedot pysyvät varmasti tallessa. Ongelmatilanteen sattuessa SQL Azure tarjoaa automaattisen vikasietoisuuden taaten sovellukselle saatavuuden. Siirtyessään SQL Azureen yrityksen ei tarvitse enää itse varmuuskopioida, säilyttää ja turvata tietoja. (Lee ym. 2009, 5.)

3.4.2 Skaalautuvuus

Yksi pääeduista jonka pilvi-malli tuo, on ratkaisujen skaalautuvuuden helppous. SQL Azurella voidaan luoda ratkaisuja, jotka vastaavat skaalautuvuusvaatimuksia, oli kyseessä sitten pieni sovellus tai maailmanlaajuinen web-palvelu. (Lee ym. 2009, 5.)

Pay-as-you-grow -hinnoittelumallilla voidaan nopeasti hankkia uusia tietokantoja tai vähentää niitä. Ei tarvitse siis maksaa käyttämättömästä kapasiteetista. (Lee ym. 2009, 5.)

SQL Azure toimii useissa tietokeskuksissa maailmanlaajuisesti. Uusia markkina-alueita voidaan tavoittaa nopeasti. Palvelu voidaan haluttaessa kohdistaa tietylle alueelle ja tietokannat voidaan luoda lähimpänä sijaitsevaan tietokeskukseen. Palvelulle saadaan maailmanlaajuinen kantama ilman

lisäkustannuksia infrastruktuurin rakentamisesta ja siihen liittyvistä hallinnollisista asioista. (Lee ym. 2009, 5.)

3.4.3 Ohjelmistokehitys

Yksi haaste hyvän pilvi-pohjaisen sovelluksen tekemiseen on uusien työvälineiden ja tekniikoiden opetteleminen. SQL Azure on rakennettu TSQL-kielen päälle ja suunniteltu yhteensopivaksi SQL Serverin kanssa ja se vaatii vain pieniä muutoksia. Tästä johtuen kehittäjät voivat hyödyntää jo valmiiksi opittua osaamista. Tämä vähentää kustannuksia ja aikaa, joka syntyy uusien tekniikoiden opettelemisesta. (Lee ym. 2009, 6.)

TSQL (Transact-SQL) tuo lisää ominaisuuksia SQL:ään. TSQL:ää käytetään SQL-palvelimien kanssa. (SearchSQLServer 2002.)

3.5 Windows Azure AppFabric

Windows Azure AppFabric tarjoaa toiminnallisuksia sovelluksille. Sovelluskehittäjät voivat hyödyntää erilaisia infrastruktuureita ja AppFabric tarjoaa muutamia osia. AppFabricin tarjoamia palveluita ovat

- palveluväylä (Service Bus)
- pääsynhallinta (Access Control)
- välimuistipalvelu (Caching). (Chappell, D. 2010, 19.)

AppFabricin osat nopeuttavat kehittämisprosessia, koska silloin ei itse tarvitse ohjelmoida samankaltaisia ominaisuuksia (Chappell, D. 2010, 19).

3.5.1 Palveluväylä

Yleinen ongelma kehitettäessä vertaisverkkosovelluksia on yhteyksien mahdollistaminen sovelluksien välillä erilaisten verkkolaitteiden, kuten palomuurien ja NAT:n läpi (Channel 9 2010).

Tämän tyyppiset verkkolaitteet tyypillisesti sallivat sovellusten yhdistää ulosmeneviä verkkoyhteyksiä, mutta estävät niitä hyväksymästä sisääntulevia yhteyksiä (Channel 9 2010).

Yhteyksien luomiseksi tällaisten verkkolaitteiden läpi sovelluskehittäjät usein joutuvat kirjoittamaan oman logiikan, joka hoitaa yhteyksien luomisen. Tämä on kuitenkin usein aika monimutkaista. (Channel 9 2010.)

Palveluväylä tarjoaa kommunikaatioinfrastruktuurin, jonka vuoksi sovelluskehittäjän ei tarvitse itse luoda monimutkaista koodia yhteyden luomiseen. Sen ansiosta pystytään paljastamaan palvelu internetissä palomuurin ja NAT:n takaa. (Channel 9 2010.)

Palveluväylä

- on rakennettu Windows Communication Foundationin (WCF) päälle ja se käyttää standardeja internet-protokollia
- on pilvipohjainen ja se on tehty helpottamaan yhteyden luomista palvelujen ja asiakkaiden välille, vaikka palvelimet ja asiakkaat olisivat palomuurien ja NAT:n takana
- ei vaadi suurta ponnistusta yritykseltä ja se johtaa nopeampiin sovelluksen julkaisuihin
- vähentää kuluja, koska kehittäjiä ei tarvitse itse tehdä monimutkaista koodia, jolla voidaan hoitaa samat asiat, jotka palveluväylä hoitaa. (Channel 9 2010.)

3.5.2 Pääsynhallinta

Pääsynhallinta tarjoaa helpon tavan tuottaa tunnistamis- ja pääsynhallintaa web-sovelluksille ja palveluille. Se on yhteensopiva esimerkiksi Active Directoryn ja muiden tunnistautumispalveluiden kanssa kuten Windows Live ID, Google, Yahoo! ja Facebook. (Windows Azure AppFabric 2010.)

Pääsynhallinta mahdollistaa tunnistamispäätösten saamisen sovelluksesta käsin sellaisessa muodossa, jonka sovellus ymmärtää. Sitä voidaan myös

käyttää käyttäjäoikeuksien hallintaan, mikä säästää aikaa ja vaivaa, sillä kehittäjien ei tarvitse itse kehittää vastaavanlaisia ominaisuuksia. Käyttäjien ja käyttäjäryhmien oikeudet ovat täysin räätälöitävissä. Pääsynhallintaa voi käyttää myös palveluväyläyhteyksiin. (Windows Azure AppFabric 2010.)

3.5.3 Välimuistipalvelu

Välimuistipalvelun käyttö tarjoaa sovellusvälimuistipalvelun Windows Azure- ja SQL Azure -sovelluksille. Se tarjoaa todella nopean pääsyn ja korkean saatavuuden sovelluksen tiedoille. Kyseinen toiminnallisuus on tarjottu kokonaan palveluna, eikä siihen vaadita minkäänlaista asennusta tai hallinnointia ja välimuisti suurenee ja pienenee dynaamisesti tarpeen mukaan. (Windows Azure AppFabric 2010.)

Käyttämällä välimuistia voidaan parantaa web-sovellusten suorituskykyä. Käyttöönotto vaatii vain pieniä asetusmuutoksia. Koodimuutoksia sovellukseen ei tarvitse tehdä. Välimuistin käyttö vähentää myös latenssia, kun käsitellään tietoa, joka on tallennettu SQL Azuren tietokantoihin tai Windows Azure varastoon. (Windows Azure AppFabric 2010.)

4 Amazon Web Services

Amazon Web Services on IaaS-tyyppinen alusta, joka tarjoaa internetin välityksellä infrastruktuuria eli laitteistoa. Palveluihin kuuluu muun muassa virtuaalikoneinstansseja, joista saadaan laskentatehoa ja joiden avulla voidaan ajaa omia sovelluksia ja palveluita. Muita palveluita ovat esimerkiksi tietovarastopalvelu ja palvelut, joilla voidaan valvoa pilvessä olevien resurssien käyttöä.

Amazon Web Servicen tuomien palveluiden avulla voidaan toteuttaa monentyyllisiä sovelluksia ja ratkaisuja. Amazon Web Service on joustavampi kuin Windows Azure Platform ja Google App Engine, mutta jossain tapauksissa kuitenkin saattaa olla työläämpää toteuttaa sovellus Amazon Web Servicen palveluilla, koska joudutaan itse hankkimaan tarvittavat ohjelmat, joiden avulla voidaan omaa sovellusta tarjota.

4.1 Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) on internetin kautta tarjottava palvelu, jossa tarjotaan laskentakapasiteettia käyttäjän tarpeiden mukaan. (Amazon Web Services 2010.)

Amazon EC2 -instansseista voidaan maksaa käytön mukaan tai vaihtoehtoisesti voidaan myös maksaa kiinteää hintaa. Kiinteää hintaa maksettaessa voidaan saada alennuksia. Olemassa on myös niin sanottuja ”spot-instansseja”, jossa hinta vaihtelee kysynnän ja tarjonnan mukaan.

Amazon EC2 -instanssit ovat virtuaalikoneita, joihin käyttäjä valitsee käytettäväksi joko valmiiksi tehtyjä virtuaalikoneita tai vaihtoehtoisesti tekee ne täysin itse omien tarpeiden mukaan. Uusien instanssien käynnistys vie vain muutamia minuutteja, joten käytettävä kapasiteetti on nopeasti skaalattavissa ylös tai alas.

Amazon EC2 -instansseissa on valittavana useampia instanssityyppejä, käyttöjärjestelmiä ja ohjelmistoja. Valinnat vaikuttavat instanssin hintaan.

4.1.1 Elastic Block Store

Amazon Elastic Block Store (EBS) tarjoaa kiintolevyjä, joita voidaan käyttää yhdessä Amazon EC2 -instanssien kanssa. Amazon EBS -levyt ovat Amazon EC2 -instanssien ulkopuolella, joten levyjen tieto säilyy vaikka Amazon EC2 -instanssi pysäytettäisiin. (Amazon Web Services 2010b). Toisin sanoen Amazon EBS -levyt eivät ole Amazon EC2 -instansseissa suoraan, vaan EBS-levyt sijaitsevat Amazonin palvelimilla.

Amazon EBS tarjoaa helposti saatavia ja luotettavia kiintolevyjä, jotka voidaan liittää käynnissä oleviin Amazon EC2 -instansseihin ja paljastaa asemina instanssien sisällä. (Amazon Web Services 2010b.) Käyttäjälle se ei eroa paljonkaan normaalista kiintolevystä. Amazon EBS sopii hyvin sovelluksiin, jotka tarvitsevat tietokannan, tietojärjestelmän tai kiintolevytilaa (Amazon Web Services 2010b).

4.1.2 Amazon CloudWatch

Amazon CloudWatch on palvelu, jonka avulla voidaan tarkastella pilvessä olevia resursseja. Sillä voidaan tarkastella Amazon EC2 -instanssien tilaa, esimerkiksi resurssien käyttöä, suorituskykyä ja kokonaistarvetta. Amazon CloudWatchilla nähdään statistiikkaa esimerkiksi prosessoritehon käytöstä, kovalevyn lukemisista ja kirjoituksista sekä verkkoliikenteen käytöstä. (Amazon Web Services 2010c.)

Amazon CloudWatchilla voi

- tarkastella Amazon EC2 -instansseja, Amazon EBS -levyjä, Elastic Load Balancereita ja RDS (Relational Database Service) -tietokantainstansseja reaaliaikaisesti
- dynaamisesti lisätä tai poistaa Amazon EC2 -instansseja Amazon Auto Scaling -ominaisuuden ansiosta CloudWatch-käyttötietojen perusteella. (Amazon Web Services 2010c.)

4.1.3 Auto Scaling

Auto Scaling -toiminnallisuuden avulla voidaan automaattisesti skaalata Amazon EC2 -kapasiteettia ylös tai alas määriteltyjen ehtojen mukaan. Auto Scaling pitää huolen siitä, että Amazon EC2 -instanssien määrä skaalautuu ylöspäin automaattisesti mahdollisina ruuhka-aikoina, jotta tarvittava suorituskyky säilyy. Sama toimii myös toiseen suuntaan eli silloin, kun esimerkiksi käyttäjämäärä on vähäistä, Auto Scaling skaalaa instanssien määrää alas. Tämän ansiosta tehdään säästöjä, koska maksetaan vain käytössä olevasta kapasiteetista. (Amazon Web Services 2010d.)

Auto Scalingilla voidaan

- asettaa ehtoja milloin Amazon EC2 -kapasiteetti skaalautuu ylös tai alas. Kun jokin ehdoista täyttyy, niin Auto Scaling automaattisesti tekee määritellyt toiminnot.
- skaalata sovellusta ruuhka-aikoina kun kysyntä kasvaa.
- vähentää Amazon EC2 -instanssien määrää, kun kysyntä laskee. (Amazon Web Services 2010d.)

4.1.4 Elastic Load Balancing

Elastic Load Balancing jakaa automaattisesti verkkoliikennettä Amazon EC2 -instanssien välillä. Sen avulla voidaan saavuttaa parempi virheensietokyky sovelluksille. Jos yksittäistä instanssia kuormittaa liikaa, niin sovelluksen suorituskyky voi laskea. Elastic Load Balancing havaitsee vialliset instanssit, jotka eivät jostain syystä toimi niin kuin niiden pitäisi ja automaattisesti ohjaa liikennettä toimiviin instansseihin, kunnes vika on korjattu. (Amazon Web Services 2010e.)

Elastic Load Balancer voidaan asettaa toimimaan yksittäisellä saatavuusalueella, tai myös useammalla alueella, jolloin saavutetaan entistä parempi toimivuus. (Amazon Web Services 2010e.) Jos esimerkiksi ensisijaisella saatavuusalueella kaikki instanssit ovat kovassa kuormituksessa, voidaan liikennettä ohjata muilla alueilla sijaitseviin instansseihin.

4.1.5 Korkean suorituskyvyn laskenta

Yrityksillä voi olla vaikeuksia ratkaista ongelmia, jotka vaativat suurien tietomäärien prosessointia. Näiden ongelmien ratkaiseminen voi olla vaikeaa, koska yrityksen oma infrastruktuuri tai budjetti ei välttämättä riitä kyseisen ongelman ratkaisemiseen. Tällaisissa tapauksissa esimerkiksi projektin aikataulu ei riitä siihen, että ongelmaa yritettäisiin ratkoa omilla laitteilla. (Amazon Web Services 2010h.)

Asioita monimutkaistaa myös muun muassa budjetointi, pääoman käyttö ja olemassa olevien resurssien priorisointi eri projektien välillä. Tällaiset haasteet usein johtavat siihen, että laskennallisia resursseja ei ole riittävästi tai olemassa olevia kalliita resursseja jää käyttämättä. (Amazon Web Services 2010h.)

Käyttämällä Amazonin EC2:ta, tällaisten suurten laskennallisten ongelmien ratkaiseminen voi helpottua huomattavasti. Amazon EC2 tarjoaa laskennallisia resursseja joustavasti ja kustannustehokkaasti.

Sen avulla yritykset voivat

- vähentää kustannuksia ja poistaa tarpeen hankkia, asentaa ja hoitaa kalliita yrityksen sisäisiä laitteistoja
- kokeilla uusia innovaatioita nopeasti, koska saatavilla on laskennallista tehoa, jonka saa käyttöön minuuteissa
- skaalata oikean määrän laskentaresursseja jokaista työtä kohden
- välttää aiheutuvia kuluja laskentatehosta, jota ei enää tarvitakaan
- muuntaa kalliit ja joustamattomat yrityksen sisäiset hallinnointikulut joustaviksi tuntipohjaisiksi kuluiksi. (Amazon Web Services 2010h.)

4.2 Amazon S3

Amazon Simple Storage Service (Amazon S3) on pilvessä sijaitseva varastopalvelu. Amazon S3 tarjoaa yksinkertaisen rajapinnan, jonka avulla voidaan tallentaa ja hakea tietoa paikasta riippumatta, kunhan käyttäjällä on pääsy internetiin. Se tarjoaa kehittäjälle samanlaisen skaalautuvan, luotettavan,

turvallisen ja nopean infrastruktuurin, jota Amazon käyttää omissa maailmanlaajuisissa web-sivustoissaan. (Amazon Web Services 2010g.)

4.2.1 Ominaisuuksia

Amazon S3:ssa käyttäjä luo säiliön, jonne tietoa tallennetaan. Säiliö voidaan tallettaa lähimpänä sijaitsevalle alueelle, jonka ansiosta voidaan minimoida latenssia. Tiedon säilyvyys taataan Amazon S3 SLA:ssa, joten tieto on hyvin saatavilla. (Amazon Web Services 2010g.)

Amazon S3:een kuuluu seuraavanlaisia ominaisuuksia:

- 1 bitin - 5 gigatavun kokoisia tiedostoja voidaan kirjoittaa, lukea ja poistaa. Tiedon määrä varastossa on rajaton.
- Kaikki tieto varastoidaan säiliöön ja haetaan uniikilla kehittäjän määrittämällä avaimella.
- Säiliö voidaan säilöä eri alueilla. Alueen valinnassa kannattaa ottaa huomioon esimerkiksi latenssi tai hinta. Amazon S3 on saatavilla Amerikassa, Euroopassa ja Aasian ja Tyynenmeren alueilla.
- Alueelle tallennetut tiedot eivät koskaan lähde kyseiseltä alueelta, ellei niitä itse siirretä. Esimerkiksi Euroopan alueella sijaitsevat tiedot eivät lähde kyseiseltä alueelta.
- Käyttäjätunnistamiseen on tarjottu mekanismit, jotta tieto voidaan pitää salassa luvattomilta käyttäjiltä. Tiedot voidaan tehdä yksityisiksi tai julkisiksi, ja niihin voidaan myöntää oikeuksia tietyille käyttäjille.
- Käyttää standardeihin pohjautuvia REST- ja SOAP-rajapintoja.
- Lataus (download) -protokollana käytetään HTTP:tä. Myös BitTorrent-protokollalle tarjotaan rajapinta.
- Luotettavuus pohjautuu Amazon S3 SLA:han, jossa kuukaudelle ilmoitettu saatavuustaso on 99,9%. (Amazon Web Services 2010g.)

4.2.2 Tiedon suojaus

Amazon S3:ssa objektit eli toisin sanoen tieto, on redundantisti varastoitu usealle laitteelle ja useisiin tiloihin kyseisellä Amazon S3 -alueella. Tiedon säilyvyyden turvaamista tehostetaan sillä, että tietoa kopioidaan useasti. Amazon S3 havaitsee, jos tietoa on menetetty, ja korvaa menetetyn tiedon talteen otetuista kopioista. Tiedon eheys tarkistetaan tietyin väliajoin käyttäen tarkistussummia. Jos viallisia tiedostoja löytyy, ne korjataan käyttäen redundantteja kopioita. Kaikkea verkon kautta siirtyvää liikennettä tarkastellaan viallisen tiedon varalta, silloin kun tietoa tallennetaan tai haetaan. (Amazon Web Services 2010g.)

Amazon S3 tarjoaa lisäsuojauksia myös versioinnin myötä. Versioinnin avulla voidaan säilyttää, hakea ja palauttaa aiempia versioita talletetusta tiedosta. Tämän avulla voidaan nopeasti palauttaa esimerkiksi tahattomasta käyttäjän tekemästä virheestä tai sovelluksen aiheuttamasta viasta rikkoutunut tiedosto. Vakiona ohjelma hakee viimeisimmän version tiedostosta. Vanhempi versio tiedostoista voidaan hakea määrittelemällä versio haun yhteydessä. Mitä enemmän versioita halutaan säilyttää, sitä enemmän joudutaan maksamaan, koska säilytystilan tarve kasvaa. (Amazon Web Services 2010g.)

Voidaan myös valita käytettäväksi pelkistetty redundanssivarasto (Reduced Redundancy Storage - RRS). RRS:ää käyttämällä asiakas voi säästää kustannuksissa, koska vähemmän tärkeitä tietoja säilytetään pienemmällä redundanssilla kuin standardissa Amazon S3 -varastossa. (Amazon Web Services 2010g.)

5 Google App Engine

Google App Engine on PaaS-mallia noudattava pilvialusta, jonka avulla voidaan ajaa sovelluksia Googlen infrastruktuurin päällä. App Engine -sovellukset skaalautuvat automaattisesti kun sovelluksen liikenne ja kapasiteettitarve kasvaa. (Google App Engine 2010a.)

Sovellusta voidaan tarjota omasta domainista (esimerkiksi www.yritys.fi) käyttämällä Google Appsiä. Vaihtoehtoisesti voidaan tarjota sovellusta käyttämällä vapaata nimeä appspot.com domainissa. Sovellukseen voidaan antaa pääsy kaikille tai pääsyyä voidaan rajoittaa vaikka jonkin tietyn organisaation henkilöille. (Google App Engine 2010a.)

Google App Engine tukee muutamilla eri ohjelmointikielillä ohjelmoituja sovelluksia. App Enginen JRE:llä (Java Runtime Environment) voi tehdä sovelluksia käyttäen standardeja Java-teknologioita, kuten JVM (Java Virtual Machine), Java-servlettejä ja Java-ohjelmointikieltä. Voidaan käyttää myös muita kieliä, jotka käyttävät JVM-pohjaista tulkkiä tai kääntäjää, kuten JavaScript tai Ruby. (Google App Engine 2010a.)

App Enginessä on myös Python-ajoympäristö, joka sisältää nopean Python-tulkin ja Pythonin standardit kirjastot. JRE ja Python-ajoympäristö on tehty varmistamaan, että sovellus toimii nopeasti, turvallisesti ja ilman, että muut sovellukset järjestelmässä häiritsevät sovelluksen toimintaa. (Google App Engine 2010a.)

App Enginestä maksetaan vain käytettyjen resurssien mukaan. Siinä ei ole asennusmaksuja eikä toistuvia maksuja. Käytetyt resurssit, kuten levytila ja kaistankäyttö lasketaan gigatavuina ja laskutus tapahtuu sen perusteella. Sovelluksien resurssien käyttöä voidaan rajoittaa asetuksilla, jonka ansiosta pysytään sille varatussa budjetissa. (Google App Engine 2010a.)

App Enginen käytön aloitus ei maksa mitään ja se on ilmainen tiettyyn rajaan asti. Kaikki sovellukset voivat käyttää maksimissaan 500 megatavua kiintolevytilaa ja sen verran prosessoritehoa ja kaistaa, että sillä pystyy

tuottamaan sovelluksen, jolla on keskimäärin 5 miljoonaa sivulatausta kuukaudessa. Näiden rajojen ylittyessä sovelluksen ylläpito alkaa maksaa. Kyseiset rajat eli kiintiöt nollaantuvat kuukausittain. Tämä tarkoittaa sitä, että pienen sovelluksen pitäminen voi pysyä täysin ilmaisena. (Google App Engine 2010a.)

5.1 Sovellusympäristö

Google App Engine sisältää seuraavanlaisia ominaisuuksia:

- Dynaaminen web-palvelu, joka tukee yleisiä web-teknologioita.
- Pysyvä tallennustila, johon voi suorittaa kyselyitä, lajitella tuloksia ja tehdä transaktioita.
- Automaattinen skaalautuvuus ja kuormantasaus.
- API:t käyttäjien varmentamiseen ja sähköpostin lähettämiseen käyttäen Googlen tilejä.
- Lokaali kehitysympäristö, joka simuloi Google App Enginea.
- Tehtäväjonot tehtävien suorittamiseen.
- Ajoitetut tehtävät tapahtumien suorittamiseen määritettyinä aikoina säännöllisin väliajoin. (Google App Engine 2010a.)

Sovellus voidaan ajaa kahdella eri ympäristöllä: Javalla ja Pythonilla. Molemmat ympäristöt tarjoavat standardit protokollat ja yleiset teknologiat web-sovellusten kehittämiseen. (Google App Engine 2010a.)

5.1.1 Hiekkalaatikko

Hiekkalaatikossa voidaan ajaa sovelluksia turvallisessa ympäristössä, jolla on rajoitetut oikeudet taustalla toimivaan käyttöjärjestelmään. Sovellukset suoritetaan omissa hiekkalaatikoissaan pilvessä. Hiekkalaatikko eristää sovelluksen omaan ympäristöönsä, joka on riippumaton laitteistosta, käyttöjärjestelmästä ja web-palvelimen fyysisestä sijainnista. (Google App Engine 2010a.)

5.1.2 Java-ympäristö

Sovelluksia voidaan kehittää JRE:lle käyttämällä yleisiä Javan web-kehitystyökaluja ja API-standardeja. Sovellukset keskustelevat käyttäen Java Servlet -standardia. Sovellus voi käyttää web-sovellustekniikoita, kuten JavaServer Pages. JRE käyttää Java 6:sta. App Engine Java SDK tukee sovellusten kehitystä Java 5:llä tai Java 6:lla. (Google App Engine 2010a.)

5.1.3 Python-ympäristö

App Enginen Python-ajoympäristöllä voidaan toteuttaa sovelluksia käyttäen Python-ohjelmointikieltä. App Engine sisältää API:n ja työkalut Python-pohjaisten web-sovellusten kehittämiseen. Sillä pystyy myös hyödyntämään laajaa valikoimaa kirjastoja ja rajapintoja, kuten esimerkiksi Djangoa. (Google App Engine 2010a.)

5.1.4 Datastore

App Engine tarjoaa tietovarastopalvelun, johon voidaan suorittaa kyselyitä ja transaktioita. Samalla tavalla kuin web-palvelin kasvaa liikenteen mukaan, myös tietovarasto kasvaa tiedon lisääntyessä. (Google App Engine 2010a.)

App Enginen datastore on erilainen kuin perinteiset relaatiotietokannat. Data-objektit, tai entiteetit, omaavat tyypin ja joukon ominaisuuksia. Kyselyt voivat noutaa entiteettejä annetun tyypin mukaan suodatettuina ja järjestettyinä ominaisuuksien arvojen mukaan. (Google App Engine 2010a.)

Datastoren entiteetit ovat skeemattomia, eli niillä ei ole määrättyä rakennetta. Entiteettien rakenne määritellään käyttäjän omassa ohjelmakoodissa. Javan JDO/JPA-rajapinnat ja Pythonin datastore-rajapinta sisältävät ominaisuuksia rakenteen käyttämiseen ja valvomiseen sovelluksessa. (Google App Engine 2010a.)

Datastore käyttää optimistic concurrency controllia (OCC). OCC varmistaa ennen commit-toimintoa, että transaktiot eivät ole muuttaneet toistensa tietoja.

Jos ristiriitoja löytyy, suoritetaan rollback-toiminto, joka palauttaa tiedot aiempaan tilaansa. OCC:ta käyttämällä tietoja ei tarvitse lukita siksi aikaa, kun transaktioita tehdään, mikä puolestaan nopeuttaa järjestelmää. Toisaalta, jos ristiriitoja tapahtuu usein, suorituskyky saattaa hidastua huomattavasti. (MSDN 2011.)

Sovellus pystyy suorittamaan useita datastore-operaatioita yhdessä transaktiossa, jossa joko kaikki operaatiot onnistuvat tai epäonnistuvat. Tällä taataan se, että tieto pysyy eheänä. (Google App Engine 2010a.)

5.1.5 Google-tilit

App Engine tukee sovellusten integraatiota Google-tilihin käyttäjän tunnistamista varten. Sovellus voi sallia käyttäjän kirjautumisen Google-tilillä, josta voidaan myös päästä käsiksi sähköpostiosoitteeseen ja käyttäjänimeen, joka tilille on annettu. Googlen tilillä käyttäjä pääsee nopeammin käsiksi sovellukseen, koska ei tarvitse luoda uutta käyttäjätiliä. Sovelluskehittäjien ei tarvitse myöskään luoda uutta tilijärjestelmää sovellusta varten. (Google App Engine 2010a.)

Users API mahdollistaa sen, että voidaan tarkistaa, onko kyseinen käyttäjä sovelluksen pääkäyttäjä. Tämä mahdollistaa "vain pääkäyttäjille" -alueiden toteuttamisen helposti sovelluksille. (Google App Engine 2010a.)

5.1.6 App Engine -palvelut

App Engine tarjoaa erilaisia palveluita, joiden avulla pystytään suorittamaan yleisiä operaatioita. Edellä olevat API:t on tarjottu kyseisille toiminnoille:

URL Fetch

Sovellukset pääsevät käsiksi internetissä oleviin resursseihin, kuten web-palveluihin tai muuhun dataan, käyttäen App Enginen URL fetch-palvelua. URL fetch-palvelu noutaa web-resursseja käyttäen Googlen nopeaa infrastruktuuria, joka on käytössä monessa Googlen tuotteessa. (Google App Engine 2010a.)

Sähköposti

Sovellukset voivat lähettää sähköpostiviestejä käyttäen App Enginen sähköpostipalvelua (Google App Engine 2010a).

Memcache

Memcache-palvelu tarjoaa sovellukselle tehokkaan muistinsisäisen avain-arvo-tyyppisen välimuistin, johon pääsee käsiksi useammasta sovelluksen instanssista. Memcache on hyödyllinen tiedolle, jota ei ole tarvetta säilyttää pitkään tai joka ei tarvitse datastoren transaktionaalisia ominaisuuksia, kuten väliaikaistiedot tai tiedot, jotka on kopioitu datastoresta välimuistiin nopeaa hakua varten. (Google App Engine 2010a.)

Kuvamuokkaus

Kuvapalvelu mahdollistaa kuvien muokkaamisen sovelluksessa. Sillä pystyy muuttamaan kuvan kokoa, rajaamaan, kierittämään ja kääntämään kuvia JPEG- ja PNG-formaateissa. (Google App Engine 2010a.)

Ajastetut tehtävät ja tehtäväjonot

Web-pyyntöihin vastaamisen lisäksi sovellus voi suorittaa myös muita tehtäviä. Tehtäviä voidaan suorittaa määriteltujen aikataulujen mukaan, esimerkiksi päivittäin tai tunneittain. Sovellus voi myös itse asettaa tehtäviä jonoon ja suorittaa niitä ajoitetusti, kuten esimerkiksi taustatehtävä joka on luotu web-pyyntöä käsiteltäessä. (Google App Engine 2010a.) Taustatehtävä voi olla esimerkiksi tehtävä, joka ajastimen avulla suorittaa sovelluksessa jonkin tehtävän.

Tehtäväjonot on tällä hetkellä julkaistu vain kokeellisena toimintona. Toistaiseksi vain Python runtime environment pystyy käyttämään tehtäväjonoja. Java-sovelluksille tehtäväjono-rajapinta julkaistaan lähitulevaisuudessa (tilanne 30.1.2011). (Google App Engine 2010a.)

5.2 Ohjelmistokehitys

Javalla ja Pythonilla oleva App Engine SDK sisältää web-palvelinsovelluksen, joka jäljittelee kaikkia App Enginen palveluita lokaalilla koneella. Molemmat SDK:t sisältävät kaikki API:t ja kirjastot, jotka ovat saatavilla App Enginessä. Web-palvelin simuloi myös hiekkalaatikkoympäristön. (Google App Engine 2010a.) Käytännössä se siis sisältää testiympäristön sovellukselle, jolla voidaan testata sovelluksen toimivuutta ja App Enginen tarjoamia palveluita ennen kuin se julkaistaan muiden käyttöön.

Molemmissa SDK:issa on työkalu, jonka avulla voidaan siirtää sovellus App Engineen. Ohjelmistokoodin ja asetustiedostojen teon jälkeen voidaan sovellus siirtää App Engineen kyseistä työkalua käyttämällä. Siirron yhteydessä työkalu kysyy Google-tilin sähköpostiosoitetta ja salasanaa. (Google App Engine 2010a.)

Kun App Enginessä olevasta sovelluksesta kehitetään uusi versio, se voidaan lähettää App Engineen ilman, että se tulee vielä käyttäjien käyttöön. Sovelluksen vanha versio toimii taustalla, kunnes sovellus asetetaan toimimaan uuden version kautta. Uutta versiota voidaan myös testaila, vaikka sovelluksen vanha versio olisi vielä käytössä. (Google App Engine 2010a.)

Toisin sanoen sovellusta voidaan testata käytännössä, ennen kuin käyttäjät ohjataan käyttämään uusinta versiota sovelluksesta. Tällä tavalla voidaan minimoida se riski, että sovellus ei jostain syystä toimi julkaisun jälkeen.

Java SDK toimii kaikilla alustoilla, joissa on Java 5 tai Java 6. SDK on saatavilla zip-tiedostona. Käyttämällä Eclipse-kehitysalustaa, voidaan käyttää siihen asennettavaa Google-liitännäistä, jolla voidaan luoda, testata ja siirtää sovelluksia App Engineen. SDK sisältää myös komentorivi-työkalut kehityspalvelimen ajamiselle ja sovelluksen siirtämiselle. (Google App Engine 2010a.)

Python SDK toimii kaikilla alustoilla, joissa on Python 2.5. SDK on saatavilla zip-tiedostona. Asennustiedostot ovat saatavilla Windowsille ja Mac OS X:lle. (Google App Engine 2010a.)

Administration Console on web-pohjainen käyttöliittymä App Enginessä toimivien sovelluksien hallinnoimiseen. Sillä voidaan luoda uusia sovelluksia, muokata domain-nimiä, muuttaa käytettävää sovelluksen versiota, tutkia lokeja ja selata sovelluksen datastorea. (Google App Engine 2010a.)

5.3 Kiintiöt ja rajoitukset

App Engine -sovelluksen luominen on ilmaista. Tilin luominen ja sovelluksen julkaisu on ilmaista, eikä se sisällä velvoitteita. Ilmaisella käyttäjätillillä sovellus saa käyttöönsä 500 mb kiintolevytilaa ja korkeintaan 5 miljoonaa sivun katselua kuukaudessa. Kun tarpeet ylittävät ilmaisen tilin rajat, laskutuksella saadaan käyttöön lisää resursseja. Laskutuksessa voidaan asettaa päivän maksimibudjetti ja allokoida budjetti jokaiselle resurssille erikseen. (Google App Engine 2010a.)

Esimerkiksi voidaan asettaa maksimimäärä tiedon tallentamiselle. Kun raja saavutetaan niin annetaan virheilmoitus. Muita resursseja, joille voidaan asettaa rajoja, ovat muun muassa sähköpostien lähetys, lähtevän ja tulevan kaistankäyttö ja prosessoritehon käyttö. Jokaista kehittäjätiliä kohden voi olla korkeintaan 10 sovellusta rekisteröitynä (Google App Engine 2010a.)

Jokainen sovellus toimii myönnettyjen resurssien rajoissa tai kiintiössä (quota). Kiintiö määrittelee, kuinka paljon annettua resurssia sovellus voi käyttää kalenteripäivän aikana. Kiintiöt nollaantuvat päivittäin ja laskeminen alkaa alusta. (Google App Engine 2010a.)

Jotkin ominaisuudet asettavat rajoituksia järjestelmän vakauden suojelemiseksi. Esimerkiksi kun sovellukselle tehdään kutsu, sen täytyy vastata 30 sekunnin kuluessa. Jos sovelluksen vastaaminen kestää liian kauan, prosessi lopetetaan ja palvelin palauttaa käyttäjälle virheen. (Google App Engine 2010a.)

Sovelluksille on myös olemassa kiinteitä rajoituksia, joihin ei itse voida vaikuttaa. Tällaisilla rajoituksilla turvataan se, että jokin sovellus ei estä tai haittaa toisten sovelluksien toimintaa. Kun sovelluksen laskutus aloitetaan, niin kiinteän kiintiön rajat nousevat.

Kiintiön kiertäminen tai muu väärinkäyttö, kuten esimerkiksi saman sovelluksen ajaminen useista käyttätileistä, jotka toimivat yhdessä, on sääntöjen vastaista. Väärinkäytön seurauksena voi olla sovelluksen tai käyttäjätilin sulkeminen. (Google App Engine 2010a.)

6 Pilviteknologia-alustojen vertailu

6.1 Hinnoittelu

Hinnoittelut eri pilvialustoissa eroavat todella paljon toisistaan ja eri alustoilla on erilaisia tapoja hinnoitella tarjontaansa. Kapasiteettia voidaan muun muassa ostaa kiinteään hintaan tai siitä voidaan maksaa käytetyn kapasiteetin mukaan.

Google App Enginessä hinnoittelu on tehty yksinkertaiseksi ja erittäin toimivaksi ja tehokkaaksi. App Enginestä maksetaan vain käytetystä kapasiteetista. Esimerkiksi jos sovellus toimii App Enginessä ja sillä ei ole käyttäjiä, toisin sanoen verkkoliikennettä, niin käyttämättömästä kapasiteetista ei makseta. Tämä poikkeaa Windows Azuresta ja Amazon Web Servicestä, joissa joudutaan maksamaan, vaikka sovelluksella ei olisi yhtään käyttäjää. Windows Azuressa ja Amazon Web Servicessä kapasiteetista maksetaan tuntikohtaisesti ja vajaatunneista maksetaan aina koko tunnin hinta.

Koska erilaisia hinnoittelutapoja on todella paljon, tutkitaan vain hinnoittelua, joka tapahtuu käytön mukaan. Käytönmukaisen hinnoittelun tutkiminen on valittu, koska se on yksi pilvilaskennan avainkohdista. Hintatiedot saattavat muuttua ajan myötä. Hinnat on tarkistettu 8.2.2011.

6.1.1 Windows Azure Platform

Windows Azure Platformin käytön mukaiseen hinnoitteluun sisältyvät instanssityypit, varastotilan käyttö, tiedonsiirto, SQL Azure tietokannat, pääsynhallintatransaktiot ja palveluväyläyhteydet.

Windows Azure

Taulukossa 1 on esitetty Windows Azuren eri instanssivaihtoehtojen tuntipohjaiset hinnat. Taulukossa 1 on kerrottu myös Windows Azure varaston ja sisällönjakeluverkon hinnoittelusta.

Taulukko 1. Windows Azuren hinnoittelu (Windows Azure Platform Offer 2010)

Laskenta	
Extra small	0,05 \$ / h
Small (vakio)	0,12 \$ / h
Medium	0,24 \$ / h
Large	0,48 \$ / h
Extra large	0,96 \$ / h
Levytila	
Tallennettua GB:a kohden	0,15 \$ / kk
10,000 varastotransaktiota kohden	0,01 \$
Sisällönjakeluverkko (CDN)	
Siirrettyä GB:a kohden Euroopan ja Pohjois-Amerikan tietokeskuksista	0,15 \$
Siirrettyä GB:a kohden muista tietokeskuksista	0,20 \$
10,000 transaktiota kohden	0,01 \$

SQL Azure

Taulukossa 2 on kerrottu SQL Azure tietokantojen hinnoittelusta. Tietokantavaihtoehtoina on Web Edition ja Business Edition. Molemmissa tietokantavaihtoehtoissa hinnoittelu perustuu tietokannan kokoon ja siitä maksetaan tietty summa kuukaudessa.

Taulukko 2. SQL Azuren hinnoittelu (Windows Azure Platform Offer 2010)

Web Edition	
korkeintaan 1 GB:n kokoinen tietokanta	9,99 \$ / kk
korkeintaan 5 GB:n kokoinen tietokanta	49,95 \$ / kk
Business Edition	
korkeintaan 10 GB:n kokoinen tietokanta	99,99 \$ / kk
korkeintaan 20 GB:n kokoinen tietokanta	199,98 \$ / kk
korkeintaan 30 GB:n kokoinen tietokanta	299,97 \$ / kk
korkeintaan 40 GB:n kokoinen tietokanta	399,96 \$ / kk
korkeintaan 50 GB:n kokoinen tietokanta	499,95 \$ / kk

Windows Azure AppFabric

Taulukossa 3 on kerrottu Windows Azure AppFabriciin sisältyviin pääsynhallinnan ja palveluväylän hinnoittelusta. Palveluväylän yhteyspaketeista veloitetaan päivittäin

Taulukko 3. Windows Azure AppFabric hinnoittelu (Windows Azure Platform Offer 2010)

Pääsynhallinta	
100,000 transaktiota kohden	1,99 \$
Palveluväylä	
"pay-as-you-go" -mallin mukaan	3,99 \$ / yhteys / kk
5:n yhteyden paketti	9,95 \$ / kk
25:n yhteyden paketti	49,75 \$ / kk
100:n yhteyden paketti	199,00 \$ / kk
500:n yhteyden paketti	995,00 \$ /kk

Tiedonsiirto

Taulukossa 4 on kerrottu Windows Azure:n tiedonsiirrosta koituvista kustannuksista. Kustannukset vaihtelevat sen mukaan, millä alueella tiedonsiirto tapahtuu.

Taulukko 4. Windows Azure:n tiedonsiirron hinnoittelu (Windows Azure Platform Offer 2010)

	Pohjois-Amerikan ja Euroopan alueet	Aasian alue
Sisääntulevaa GB:a kohden	0,10 \$	0,10 \$
Ulosmenevää GB:a kohden	0,15 \$	0,20 \$

6.1.2 Amazon Web Services

Amazon Web Services -hinnoitteluun sisältyvät instanssityypit, tiedonsiirto, varastotilan käyttö, CloudWatch, AutoScaling, Elastic Load Balancing ja Amazon S3.

Amazon EC2 -instanssit

Taulukossa 5 on kerrottu Amazon EC2 -instanssien instanssivaihtoehtoista ja niiden hinnoista. Instansseista voidaan valita normaalitehoisia, pienitehoisia, suuren muistimäärän tai laskentatehon omaavia instansseja.

Taulukko 5. Amazon EC2 -instanssien hinnoittelu (Amazon Web Services 2010f)

Standardi instanssi	Linux/UNIX käyttö	Windows käyttö
Small (vakio)	0,095 \$ / h	0,12 \$ / h
Large	0,38 \$ / h	0,48 \$ / h
Extra Large	0,76 \$ / h	0,96 \$ / h
Mikro instanssi		
Mikro	0,025 \$ / h	0,035 \$ / h
Suuren muistin instanssi		
Extra Large	0,57 \$ / h	0,62 \$ / h
Double Extra Large	1,14 \$ / h	1,24 \$ / h
Quadruple Extra Large	2,28 \$ / h	2,48 \$ / h
Suuren laskennan instanssi		
Medium	0,19 \$ / h	0,29 \$ / h
Extra Large	0,76 \$ / h	1,16 \$ / h

Internet-tiedonsiirto

Taulukossa 6 on kerrottu Amazon EC2 -instanssien tiedonsiirrosta koituvista kustannuksista. Hinta vaihtelee alueittain.

Taulukko 6. Amazon internet-tiedonsiirron hinnoittelu (Amazon Web Services 2010f)

Sisäänpäin tuleva data	US & EU alueet	Aasian alue
Kaikki siirretty data	0,10 \$ / GB	0,10 \$ / GB
Ulospäin lähtevä data	US & EU alueet	Aasian alue
Kuukauden ensimmäinen GB	0,00 \$	0,00 \$
10 TB:n asti kuukaudessa	0,15 \$ / GB	0,19 \$ / GB
Seuraavat 40 TB kuukaudessa	0,11 \$ / GB	0,15 \$ / GB
Seuraavat 100 TB kuukaudessa	0,09 \$ / GB	0,13 \$ / GB
Yli 150 TB kuukaudessa	0,08 \$ / GB	0,12 \$ / GB

Samalla alueella sijaitsevien Amazon EC2 -instanssien ja Amazon Web Servicen palveluiden välisistä tiedonsiirroista ei peritä maksua. Elastic Load Balancerin sisäänpäin tulevasta ja ulospäin lähtevästä tiedonsiirrosta peritään 0,01 \$ / GB. (Amazon Web Services 2010f.)

Amazon Elastic Block Store

Taulukossa 7 on kerrottu Amazon EBS -levyjen hinnoittelusta. I/O (input/output) -pyyntöjä tapahtuu, kun tietoa liikkuu EBS-levyissä.

Taulukko 7. Amazon Elastic Block Storen hinnoittelu (Amazon Web Services 2010f)

Amazon EBS -levyt	
Käytettyä tilaa kohden kuukaudessa	0,11 \$ / GB
I/O-pyyntöt	0,11 \$ / 1 000 000

Amazon CloudWatch

Amazon CloudWatchin käyttö maksaa 0,015 \$ instanssi-tuntia, tai vajaatuntia kohden (Amazon Web Services 2010f). Jos seurataan esimerkiksi kahta Amazon EC2 -instanssia kahden tunnin ajan, niin maksetaan $(2 * 0,015 \$) * 2 = 0,06 \$$. Samaan lopputulokseen päästään, jos kyseisiä instansseja seurataan esimerkiksi 1,5 tuntia.

Auto Scaling

Auto Scalingin saa käyttöön Amazon CloudWatchin myötä, eikä siitä peritä lisämaksuja. Jokaista Auto Scalingin käynnistämää instanssia tarkastellaan automaattisesti ja siitä peritään Amazon CloudWatchin hinta. (Amazon Web Services 2010f.)

Elastic Load Balancing

Taulukossa 8 on kerrottu Elastic Load Balancerin hinnoittelusta. Hinnoittelu on tuntikohtaista, ja vajaista tunneista maksetaan kokonaisen tunnin hinta.

Taulukko 8. Amazon Elastic Load Balancing hinnoittelu (Amazon Web Services 2010f)

Elastic Load Balancing	
Jokaista Elastic Load Balancer-tuntia (tai vajaatuntia) kohden	0,028 \$
GB:a kohden, jonka Elastic Load Balancer käsittelee	0,008 \$

Amazon S3

Taulukossa 9 on kerrottu Amazon S3:n hinnoittelusta. Amazon S3:n käytön kustannukset vähenevät, mitä enemmän tietoa liikkuu tai mitä suurempi määrä tietoa varastoidaan. Hinnoitteluun sisältyy normaali varasto, pelkistetty redundanssivarasto, tiedonsiirto ja pyynnöt. Pyyntöjä tapahtuu kun Amazon S3 -varastosta esimerkiksi haetaan tietoa, tai kun sinne tallennetaan tietoa.

Taulukko 9. Amazon S3 hinnoittelu (Amazon Web Services 2010g)

Varasto		Pelkistetty redundanssivarasto	
Ensimmäinen 1 TB / käytettyä varastotilaa kuukaudessa	0,140 \$ / GB	Ensimmäinen 1 TB / käytettyä varastotilaa kuukaudessa	0,093 \$ / GB
Seuraavat 49 TB	0,125 \$ / GB	Seuraavat 49 TB	0,083 \$ / GB
Seuraavat 450 TB	0,110 \$ / GB	Seuraavat 450 TB	0,073 \$ / GB
Seuraavat 500 TB	0,095 \$ / GB	Seuraavat 500 TB	0,063 \$ / GB
Seuraavat 4000 TB	0,080 \$ / GB	Seuraavat 4000 TB	0,053 \$ / GB
Yli 5000 TB	0,055 \$ / GB	Yli 5000 TB	0,037 \$ / GB
Tiedonsiirto		Pyynnöt	
Kaikki sisääntuleva tiedonsiirto	0,100 \$ / GB	PUT, COPY, POST tai LIST pyynnöt	0,01 \$ / 1000 pyyntöä
Ensimmäinen 1 GB ulosmenevästä tiedonsiirrosta	0,000 \$ / GB	GET ja muut pyynnöt	0,01 \$ / 10000 pyyntöä
10 TB:iin asti ulosmenevästä tiedonsiirrosta	0,150 \$ / GB		
Seuraavat 40 TB	0,110 \$ / GB		
Seuraavat 100 TB	0,090 \$ / GB		
Yli 150 TB	0,080 \$ / GB		

6.1.3 Google App Engine

Googlen App Enginessä hinnoittelu on tehty huomattavasti yksinkertaisemmaksi, kuin esimerkiksi Windows Azuressa tai Amazon Web Servicessä. Taulukossa 10 on kerrottu Google App Enginen hinnoittelusta.

Taulukko 10. Google App Enginen hinnoittelu (Google App Engine 2010b)

Google App Engine	
Ulos menevä data	0,12 \$ / GB
Sisään tuleva data	0,10 \$ / GB
Prosessori käyttö	0,10 \$ / prosessorinkäyttötunti
Tallennettu data	0,15 \$ GB:a kohden kuukaudessa
Lähetetyt sähköpostit	0,0001 \$ / sähköpostiviesti

6.2 Ominaisuudet

Ominaisuuksia on verrattu hyvin yleisellä tasolla ja vertailuun on otettu mukaan oleellisia ominaisuuksia. Taulukossa 11 vertaillaan tutkittavana olevien pilvialustojen ominaisuuksia.

Taulukko 11. Pilvialustojen ominaisuuksia

Ominaisuus	Windows Azure	Amazon Services	Web	Google App Engine
Pilvilaskennan tyyli	PaaS	IaaS		PaaS
Ohjelmointikielien	Lähes kaikki	Lähes kaikki		Python, Java
Tietokanta	On	On		On
Varastotila	On	On		On
Skaalautuvuus	On	On		On
Kuormantasaus	On	On		On
Ilmainen käyttö pienelle sovellukselle (tiettyjen rajojen sisällä)	Ei	Ei		On
Monitorointi	On	On		On
Kehitystyökalut	On	Ei (itse asennettava)		On

Taulukosta 11 ilmenee, että alustoissa on paljon samoja ominaisuuksia. Ominaisuudet eri alustoissa on kuitenkin toteutettu hyvin eri tavoin. Suurimpia eroja alustoissa on tuetut teknologiat, ohjelmointikielet, hinnoittelu ja kehitystyökalut.

7 Tulokset ja johtopäätökset

Pilvilaskenta ja pilvipalvelut ovat hyvä vaihtoehto, kun tarvitaan erilaisia ratkaisuja, mutta ei ole varaa tai mahdollisuutta ostaa kalliita laitteistoja niiden ylläpitämiseen. Pilvilaskenta mahdollistaa infrastruktuurin ja kehitysalustojen hankinnan internetin välityksellä. Tärkeimpiä etuja pilvilaskennassa on sen skaalautuvuus, hinnoittelu ja lähes rajaton kapasiteetti.

Tutkittavana olleet järjestelmät eroavat toisistaan melkoisesti, vaikka niillä kaikilla kuitenkin pyritään enemmän tai vähemmän samaan lopputulokseen. Tästä johtuen alustojen vertaaminen suoraan toisiinsa koitui hankalaksi.

Työn tavoitteena oli selvittää tutkittavana olevien pilvialustojen ominaisuuksia ja tutkia, minkälaiseen tarkoitukseen ne soveltuvat. Pilvialustojen tutkimista hankaloitti se, että tietoa löytyi enimmäkseen vain palveluntarjoajan omilta internet-sivuilta ja tieto siellä oli osaksi mainostekstin kaltaista. Työn tavoitteet kuitenkin saavutettiin ja pilvialustat selvitettiin yleisellä tasolla.

Tutkittavana olleet alustat ovat huomattavasti laajempia, mitä tässä työssä on selvitetty ja aiheelle on varmasti paljon mahdollisuuksia jatkotutkimuksille. Jatkotutkimuksena voisi esimerkiksi tutkia yksittäistä pilvialustaa tarkemmin ja tehdä kyseisen alustan avulla jokin pilvisovellus.

Pilvialustaa valittaessa kannattaa huomioida esimerkiksi seuraavia asioita:

- Millaisilla ohjelmointikielillä halutaan sovellusta kehittää?
- Minkä tyyppisiä sovelluksia tehdään?
- Mikä on käytettävissä oleva budjetti?
- Kuinka paljon kontrollia halutaan taustalla toimivista järjestelmistä?

7.1 Windows Azure Platform

Windows Azure ei rajoita kehittäjiä käyttämään vain jotain tiettyä ohjelmointikieltä, vaan tuettuja kieliä on paljon. Se soveltuu erityisen hyvin sovelluksille, jotka pohjautuvat Microsoftin teknologioihin ja ratkaisuihin, koska Windows Azure integroituu muihin Microsoftin tuotteisiin.

Windows Azuressa sovelluksen pitäminen maksaa, vaikka sillä ei olisi yhtään käyttäjää. Pienelle vasta-aloittavalle yritykselle tämä voi olla este Windows Azuren käytössä. Suuremmalle yritykselle aloitukseen ja ohjelman ylläpitoon liittyvät kustannukset eivät välttämättä ole niin suuri tekijä.

Google App Enginen tavoin, kehittäjä kirjoittaa ohjelmakoodin, joka lähetetään palveluntarjoajan palvelimille. Sovellus asetetaan käyttämään palvelimella olevia varastoja ja siihen tehdään tarvittavat asetukset, jonka jälkeen sovellus on käyttövalmis.

7.2 Amazon Web Services

Amazon eroaa Azuresta ja Googlesta, koska siinä toimitettavana yksikkönä eivät ole sovelluksen kooditiedostot, vaan virtuaalikoneet. Tämä tekee Amazonin alustasta kaikkein joustavimman, koska virtuaalikoneille asennetaan omat sovellukset ja järjestelmät. Sen takia se ei sido asiakasta käyttämään esimerkiksi vain jotain tiettyä ohjelmointikieliä, vaan voidaan käyttää haluttuja teknologioita.

Amazon loistaa ratkaisuihissa, jotka vaativat suuren määrän prosessointitehoa. Esimerkkinä tästä on tapaus, jossa New York Times siirsi uutisarkistonsa internetiin selattavaksi. Tieto prosessoitiin käyttämällä 100:aa Amazon EC2 -instanssia ja aikaa siihen kului alle 24 tuntia.

Amazon mahdollistaa myös perinteisten sovellusten siirtämisen pilveen helposti, sillä sovellukseen ei tarvitse tehdä koodimuutoksia. Sovellus ja sen tarvitsemat ohjelmistot vain asennetaan Amazon EC2 -instanssiin, minkä jälkeen sovellus on käytettävissä.

Haittapuolena on se, että joudutaan asentamaan omat järjestelmät ja tarvittavat sovellukset. Tämä monimutkaistaa sovellusten ja ratkaisujen tekemistä. Lisäkuluja tulee hankittujen ja asennettujen ohjelmien lisenssimaksuista. Niinkuin Azuressa, Amazonissa sovelluksen pitäminen maksaa, vaikka sillä ei olisi ainuttakaan käyttäjää.

7.3 Google App Engine

Google App Enginen käyttö on alussa täysin ilmaista, ja pientä sovellusta voidaan pitää ilmaiseksi pilvessä, jos se pysyy tiettyjen rajojen sisällä. Hinnoittelu on kaikinpuolin yksinkertaisempaa kuin Windows Azuressa tai Amazon Web Servicessa. Pienille sovelluksille tämä on halvin vaihtoehto tutkittavana olevista kolmesta alustasta.

Hallinnollista puolta App Engine -sovelluksella on hyvin vähän. App Engine hoitaa automaattisesti muun muassa skaalautuvuuden ja kuormantasauksen. Sovellusten kehittäminen on Google App Enginellä yksinkertaisempaa, kuin Windows Azurella tai Amazon Web Servicellä.

Google App Enginessä laskutetaan vain käytetystä kapasiteetista. Jos sovelluksella ei ole käyttäjiä tai verkkoliikennettä, sen ylläpitäminen ei maksa. Tallennetusta tiedosta laskutetaan, jos se ylittää ilmaisen tilin rajat.

Huonoiksi puoliiksi voidaan katsoa se, että sillä voidaan kehittää sovelluksia vain Javalla tai Pythonilla. Tästä johtuen joudutaan opettelemaan uusia teknologioita ja ohjelmointikieliä, jos niitä ei ennestään osata.

LÄHTEET

- Amazon Web Services 2010a. Amazon Elastic Compute Cloud (Amazon EC2). Viitattu 10.8.2010 <http://aws.amazon.com/ec2/>.
- Amazon Web Services 2010b. Amazon Elastic Block Store (EBS). Viitattu 21.10.2010 <http://aws.amazon.com/ebs/>.
- Amazon Web Services 2010c. Amazon CloudWatch. Viitattu 21.10.2010 <http://aws.amazon.com/cloudwatch/>.
- Amazon Web Services 2010d. Auto Scaling. Viitattu 21.10.2010 <http://aws.amazon.com/autoscaling/>.
- Amazon Web Services 2010e. Elastic Load Balancing. Viitattu 21.10.2010 <http://aws.amazon.com/elasticloadbalancing/>.
- Amazon Web Services 2010f. Amazon EC2 Pricing. Viitattu 16.11.2010 <http://aws.amazon.com/ec2/pricing/>.
- Amazon Web Services 2010g. Amazon Simple Storage Service (Amazon S3). Viitattu 25.11.2010 <http://aws.amazon.com/s3/#functionality>.
- Amazon Web Services 2010h. High Performance Computing (HPC). Viitattu 2.12.2010 <http://aws.amazon.com/ec2/hpc-applications/>.
- Azure Academic Pilot 2010a. Introduction to Cloud Computing. Viitattu 10.8.2010 <http://www.azurepilot.com/page/Introduction+to+Cloud+Computing>.
- Azure Academic Pilot 2010b. Windows Azure Platform: Overview. Viitattu 7.9.2010 <http://www.azurepilot.com/page/Windows+Azure+Platform%3A+Overview>.
- Azure Academic Pilot 2010c. Windows Azure: Introduction. Viitattu 7.9.2010 <http://www.azurepilot.com/page/Windows+Azure%3A+Introduction>.
- Bhaskardeep 2010. Cloud computing - SaaS. Viitattu 27.7.2010 <http://www.c-sharpcorner.com/UploadFile/bhaskardeep/190/Default.aspx>.
- Channel 9 2010. Overview. Viitattu 8.12.2010 <http://channel9.msdn.com/Learn/Courses/Azure/ServiceBus/Introductiontothe.NETServiceBus/Overview>.
- Chappell, D. 2009. INTRODUCING THE WINDOWS AZURE PLATFORM. Viitattu 9.9.2010 <http://go.microsoft.com/fwlink/?LinkId=158011>.
- Chappell, D. 2010. INTRODUCING THE WINDOWS AZURE PLATFORM. Viitattu 8.12.2010 <http://go.microsoft.com/?linkid=9682907>
- Ciol 2010. Choosing the right cloud platform. Viitattu 15.3.2011 <http://www.ciol.com/Developer/Enterprise-Tools/Feature/Choosing-the-right-cloud-platform/134769/0/>.
- Google App Engine 2010a. What Is Google App Engine?. Viitattu 9.9.2010 <http://code.google.com/intl/fi/appengine/docs/whatisgoogleappengine.html>.
- Google App Engine 2010b. Billing and Budgeting Resources. Viitattu 10.12.2010 http://code.google.com/intl/fi-FI/appengine/docs/billing.html#Billable_Quota_Unit_Cost.

- Gens, F. 2008. Defining "Cloud Services" and "Cloud Computing". Viitattu 24.11.2010 <http://blogs.idc.com/ie/?p=190>.
- Learning Tree 2010. Comparing Paas and IaaS. Viitattu 14.3.2011 <http://cloud-computing.learningtree.com/2010/08/25/comparing-paas-and-iaas/>.
- Lee, J.; Malcolm, G. & Matthews, A. 2009. Overview of Microsoft SQL Azure Database. Viitattu 4.11.2010 <http://go.microsoft.com/?linkid=9686976>.
- Luu5 2010. SaaS-termejä. Viitattu 8.3.2011 <http://www.luu5.fi/?sivu=saas&sub=termeja>.
- MSDN 2006. Multi-Tenant Data Architecture. Viitattu 14.3.2011 <http://msdn.microsoft.com/en-us/library/aa479086.aspx>.
- MSDN 2011. Optimistic Concurrency Control. Viitattu 8.3.2011 [http://msdn.microsoft.com/en-us/library/bb190073\(v=ax.10\).aspx](http://msdn.microsoft.com/en-us/library/bb190073(v=ax.10).aspx).
- Pijanowski, K. 2009. Understanding Public Clouds: IaaS, PaaS, & SaaS. Viitattu 29.9.2010 http://www.keithpij.com/Home/tabid/36/EntryID/27/Default.aspx#_Toc231187513.
- Qrimp 2008. The Difference between IaaS and PaaS. Viitattu 14.3.2011 <http://www.qrimp.com/blog/blog.The-Difference-between-IaaS-and-PaaS.html>.
- Sanastokeskus TSK ry 2002. Tietotekniikan termitalkoot. Viitattu 19.11.2010 <http://www.tsk.fi/tsk/termitalkoot/fi/node/266>.
- SearchCloudComputing 2007. cloud computing. Viitattu 15.3.2011 <http://searchcloudcomputing.techtarget.com/definition/cloud-computing>.
- SearchCloudComputing 2009. Infrastructure as a Service (IaaS). Viitattu 8.3.2011 <http://searchcloudcomputing.techtarget.com/definition/Infrastructure-as-a-Service-IaaS>.
- SearchSQLServer 2002. T-SQL (Transact-SQL). Viitattu 8.3.2011 <http://searchsqlserver.techtarget.com/definition/T-SQL>.
- Verizon 2009. Most Clouds Obstruct Your Vision. This One Enhances It. Viitattu 1.2.2011 http://www.verizonbusiness.com/resources/newsletters/most-clouds-obstruct-your-vision_en_xg.pdf.
- WebDAM 2011. SCALABILITY IN THE CLOUD. Viitattu 14.3.2011 <http://www.webdamsolutions.com/digital-asset-management/scalability-in-the-cloud/>.
- Windows Azure Platform Offer 2010. Windows Azure Platform Consumption. Viitattu 16.11.2010 <http://www.microsoft.com/windowsazure/offers/popup/popup.aspx?lang=en&locale=fi-FI&offer=MS-AZR-0003P>.
- Windows Azure 2010a. Windows Azure Storage. Viitattu 10.12.2010 <http://www.microsoft.com/windowsazure/storage/default.aspx>.
- Windows Azure 2010b. Windows Azure Content Delivery Network (CDN). Viitattu 10.12.2010 <http://www.microsoft.com/windowsazure/cdn/default.aspx>.
- Windows Azure AppFabric 2010. Middleware Services. Viitattu 10.12.2010 <http://www.microsoft.com/en-us/appfabric/azure/middleware-services.aspx#ServiceBus>.
- Zoho Creator 2011. Platform as a Service. Viitattu 14.3.2011 <http://www.zoho.com/creator/paas.html>.