Lauri Miettinen

**ANSAINTAMALLIT ANDROID-SOVELLUKSISSA**

**ANSAINTAMALLIT ANDROID-SOVELLUKSISSA**

Lauri Miettinen

Opinnäytetyö, osa 1

26.2.2016

Tieto- ja viestintätekniikan koulutusohjelma

Oulun ammattikorkeakoulu

# SISÄLLYS

[SISÄLLYS 3](#_Toc449454821)

[1 JOHDANTO 4](#_Toc449454822)

[2 ANSAINTAMALLIEN ERITTELY 5](#_Toc449454823)

[2.1 Kertamaksusovellukset (Premium) 5](#_Toc449454824)

[2.2 Sovelluksen sisäiset ostokset (Freemium) 5](#_Toc449454825)

[2.3 Tilaukset 6](#_Toc449454826)

[2.4 Mainokset 6](#_Toc449454827)

[3 ANSAINTAMALLIEN TOTEUTUS 7](#_Toc449454828)

[3.1 GOOGLE PLAY-KAUPPA 8](#_Toc449454829)

[3.2 SOVELLUKSENSISÄISET OSTOKSET 9](#_Toc449454830)

[3.2.1 Hallinnoidut sovelluksen sisäiset ostokset 10](#_Toc449454831)

[3.2.2 Ostotapahtuman eteneminen 10](#_Toc449454832)

[3.2.3 Tilaukset 12](#_Toc449454833)

[3.3 MAINOKSET 13](#_Toc449454834)

[3.3.1 Käyttöönotto 13](#_Toc449454835)

[3.3.2 Mainosbannerit 13](#_Toc449454836)

[3.3.3 Koko näytön mainokset 14](#_Toc449454837)

[4 JOHTOPÄÄTÖKSET 15](#_Toc449454838)

[LÄHTEET 16](#_Toc449454839)

# JOHDANTO

Tässä työssä tutustutaan perusteisiin siitä, miten Android-sovelluksilla voidaan tehdä liiketoimintaa.

Maailman laajuisesti myydyistä älypuhelimista jopa 82 % pitävät sisällään Android-käyttöjärjestelmän (1). Kuluttajien suosion saavuttanut Android on myös saavuttanut sovelluskehittäjien suosion. Android-sovelluksia on helppo kehittää ja testata. Koska kehittäjien yhteisö on suuri, Android-ohjelmointiin on helppo saada apua, jos ohjelmistoyrityksellä tulee ongelmia vastaan. Nämä asiat yhdessä tekevät Androidista varteenotettavan vaihtoehdon alkavan yrityksen mobiilisovelluksen alustana.

Pelit omistavat hyvin suuren osan Android-sovelluksen markkinaosuudesta. Tämä työ, ja siinä käsitellyt esimerkit keskittyvät hyötysovelluksiin, vaikka työssä kuvattuja oppeja voidaan yhtä hyvin soveltaa myös pelisovelluksiin.

Työssä käydään läpi eri ansaintamalleja, vertaillaan niitä ja tutkitaan, minkälaisissa sovelluksissa niitä voidaan käyttää. Ansaintamalleja ja niiden hyviä ja huonoja puolia eritellään ja analysoidaan.

Työssä käsiteltävä ohjelmistotekniikka keskittyy Android-sovelluksiin. Tutkitaan, miten ansaintamallin sisältävä sovellus toteutetaan Android Studio-kehitysalustalla, ja miten toteutus tehdään Google Play -kaupan ohjelmistorajapinnoilla.

Jokaisesta työhön otettavasta ansaintamallista tehdään selvitys, minkälaista tekniikkaa ja mitä rajapintoja Google Play -kaupassa niiden toteutus vaatii. Tutkitaan, mitä työtä ohjelmistokehittäjän on tehtävä, jotta kukin ansaintamalli saadaan toteutettua.

Työssä tehdään selvitys mobiilisovellusmarkkinoista. Johtopäätöksessä tehdään yhteenveto mobiilisovellusliiketoiminnan kannattavuudesta.

# ANSAINTAMALLIEN ERITTELY

## Kertamaksusovellukset (Premium)

Yksinkertainen ansaintamalli on kertamaksusta ladattava sovellus. Android-kehittäjä tekee sovelluksen ja lähettää sen Google-kauppaan. Käyttäjä voi maksaa sovelluksesta kertamaksun, jolloin hän saa ladattua sen laitteellensa. Malli soveltuu hyvin sovelluksille, jotka ovat laajoja ja tarjoavat paljon ominaisuuksia. (2.)

Yksi kertamaksumallin huono puoli on se, että jokainen asiakas maksaa sovelluksesta vain kerran. Muut mallit, kuten sovelluksen sisäiset ostot, voivat pitkän ajan kuluessa saada aikaan enemmän tuloja asiakasta kohden. (2.)

## Sovelluksen sisäiset ostokset (Freemium)

Sovelluksen voi jakaa ilmaiseksi, mutta siten, että käyttäjät voivat vaihtoehtoisesti maksaa rahaa sovelluksen sisäisissä ostoksissa. Ostokset voivat olla ominaisuuksia, jotka ovat lukittuja ilmaisessa versiossa. Maksamalla niistä voi ominaisuuden saada käyttöön. Sovelluksensisäiset ostokset voivat myös olla kulutettavia, esimerkiksi pelinsisäisen valuutan ostaminen.

Tämä malli voi olla yritykselle hyvin antoisa. Käyttäjät, jotka ostavat sovelluksensisäisiä ostoksia, ovat hyvin kiinnostuneita ja aktiivisia. Heidän hintaherkkyytensä sovelluksensisäisiin ostoksiin on pieni, eli he ovat valmiita kuluttamaan suuriakin määriä rahaa. Koska sovelluksen peruskäyttö onnistuu maksamatta, uusi käyttäjä voi tutustua sovellukseen kuluttamatta rahaa. Tällöin ostaja tuntee sovelluksen peruskäytön, ja oston jälkeen asiakastyytyväisyys on korkeampi. (3, s. 176.)

## Tilaukset

Tilaus tarkoittaa sitä, että käyttäjä saa sovelluksen ominaisuuksia käyttöönsä, jos hän maksaa ajoittaisen tilausmaksun. Esimerkiksi uutissovelluksen tilaus voi maksaa kerran kuukaudessa.

Tämän mallin ansiosta yritykselle tulee jokaisesta tilaajasta jatkuva rahavirta, mikä voi olla hyvin tuottoisaa. Sovelluksen on oltava hyvin laadukas ja hyödyllinen, jotta tilaajat pysyisivät maksavina asiakkaina. (4.)

## Mainokset

Sovelluksen lataus ja käyttö voidaan myös tehdä täysin ilmaiseksi. Tällöin sovellukseen voidaan lisätä mainoksia. Google on kehittänyt Admob-palvelun, jolla kehittäjä voi lisätä Android-sovelluksiinsa mainosbannereita. Jokaisesta kerrasta kun mainos ladataan käyttäjän Android-sovellukseen, se lasketaan yhdeksi katselukerraksi, eli impressioksi. Mainostajan maksamasta hinnasta voidaan käyttää CPM-lukua (Cost Per Mille, missä mille tarkoittaa kreikaksi tuhatta), mikä tarkoittaa mainonnan hintaa tuhannesta katselukerrasta. Mainostajat maksavat Admobille CPM-luvun mukaisesti, saadakseen mainoksensa näkyviin palveluun. Admob maksaa palkkion kehittäjille joiden sovellusten mainokset keräävät suuriä määriä katselukertoja. (4; 5, s 393; 7)

# ANSAINTAMALLIEN TOTEUTUS

Google kehittää jatkuvasti uusia Android-käyttöjärjestelmän versioita. Jokaisessa käyttöjärjestelmäversiossa tulee uusia ominaisuuksia, joita Android-kehittäjä voi hyödyntää sovelluksia kehittäessä.

Jos liian vanhalla käyttöjärjestelmäversiolla yrittää avata sovelluksen, jonka kohde käyttöjärjestelmätaso on korkeampi, sovelluksesta saattaa puuttua ominaisuuksia. Liian vanhalla Android-käyttöjärjestelmällä sovellus voi kaatuilla jatkuvasti ja olla käyttökelvoton.

Ei myöskään ole kannattavaa suunnitella sovellusta yhteensopivaksi liian matalan ohjelmointirajapintatason kanssa. Uudemmissa tasoissa on paljon uusia funktioita ja luokkia, jotka voivat olla hyvin hyödyllisiä sovellusta kehittäessä.

Android Studio -projektia tehtäessä, kehittäjän täytyy valita sovelluksen minimi ohjelmointirajapintataso. Taso täytyy valita siten, että mahdollisimman moni sovelluskaupan käyttäjä pystyy käyttämään sovellusta, mutta kumminkin siten, että taso sisältää mahdollisimman uudet luokkakirjastot (kuva 1).

Kitkat 4.4 -käyttöjärjestelmä on yleisin Android-puhelimissa esiintyvä käyttöjärjestelmä. Tämä käyttöjärjestelmä tukee ohjelmointirajapinnan tasoa 19 ja on taaksepäin yhteensopiva kaikkien aikaisempien Android-versioiden kanssa. 70,9 % Google Play -kaupassa aktiivisista puhelimista tukevat tasoa 19. Kuvan 1 oikeassa laidassa kuvaillaan, mitä uusia ominaisuuksia ohjelmointirajapintatasoon 19 on tullut aikaisempiin tasoihin verrattuna.

Sovelluskehittäjän kannattaa tehdä sovelluksestaan yhteensopiva ohjelmointirajapinnan tason 19 kanssa. Jos kehittäjä määrittelee projektin Android-manifestissa alimman rajapintatason tasolle 19, Android Studio huomauttaa kehittäjää, jos hän vahingossa käyttää metodeja, jotka eivät ole yhteensopivia alimmalla tasolla. Myös Android-dokumentaatiossa kuvaillaan jokaisen metodin kohdalla, minkä tasoisen rajapinnan kanssa ne ovat yhteensopivia. (7.)



**Kuva 1.** Android Studio -ohjelmassa ohjelmointirajapintatason valintaa opastava kuvaaja.

## GOOGLE PLAY-KAUPPA

Google Play-kaupan käyttöönotto vaatii Google Play julkaisijatilin, sekä kauppiastilin luomista. Google Play-kauppaan pystyy lähettämään sovelluksia kuka tahansa, mutta maksullisten sovellusten tekeminen vaatii kauppiastilin. (8.)

Google Play kauppiastili vaatii käyttöön ottaessa 25 $:n rekisteröitymismaksun. Maksu on hyvin edullinen. Microsoftin mobiilikehittäjämaksu vaatii 20 $ kertamaksun. Apple vaatii kalliin, 100 $ vuodessa maksun yksityishenkilöltä, joka haluaa julkaista iOS-sovelluksia. (9; 10)

Google tarjoaa Google Play -kehityskonsolin, jota voi käyttää internet-selaimella. Konsolin kautta kehittäjä voi julkaista Android Studiolla kehittämänsä sovellukset. Sovelluksen julkaisemisen jälkeen kehittäjäkonsoli tarjoaa tilastoja siitä, millä tavoin asiakkaat käyttävät sovellusta. Kehittäjä voi julkaista palautteen avulla päivityksiä sovellukseensa. Lisäksi tilastot voivat auttaa kehittäjää suunnittelemaan tulevia sovelluksiaan.

Kun sovellus on valmis, kehittäjä lataa sovelluspaketin (APK, Application Package) Google Play -kehittäjäkonsoliin. Kehittäjä lataa valmistellut kuvakaappaukset, videot sekä sovelluksen kuvaustekstit. Kauppasivun on vakuutettava asiakas sovelluksen nimellä, kuvaustekstillä, esittelyvideolla ja kuvakaappauksilla. Google Playssa kävijät voivat arvostella lataamiaan sovelluksia. Tyytyväiset käyttäjät jättävät positiivisia arvosteluja, houkuttaen yhä useammat käyttäjät lataamaan sovelluksen, mikä tarkoittaa kehittäjälle yhä enemmän maksavia asiakkaita.

Ennen julkaisua kehittäjä määrittelee, onko sovellus ilmainen vai maksullinen, ja määrittää sovelluksen hinnan. Sovelluksen julkaiseminen vaatii kehittäjältä perinpohjaista testaamista. (8.)

Google Play-kauppa ei suinkaan ole ainoa vaihtoehto sovelluksen julkaisemiselle. On myös mahdollista julkaista Android-sovellus omalla verkkosivuilla tai julkaista se jossakin toisessa internetissä olevassa sovelluskaupassa. Esimerkiksi jos ohjelmistoyritys haluaa julkaista sovelluksen ainoastaan ammattilaiskäyttöön, laajaan levitykseen suunniteltu Google Play ei ehkä ole ihanteellinen julkaisualusta. (11.)

## SOVELLUKSENSISÄISET OSTOKSET

Kun sovelluksessa pyritään tekemään ostos, sovellus kommunikoi puhelimessa olevan Google Play -sovelluksen kanssa. Jos käyttäjä on tallentanut maksukeinon Google Play -sovellukseen, sovellus käsittelee ostopyynnön ja välittää tiedon ostoksen onnistumisesta takaisin sovellukselle jossa osto tehtiin. Tällöin käyttäjä näkee, kuinka ostettu tuote ilmestyy sovellukseen käytettäväksi. (12.)

Järjestelmässä on monia etuja. Yksi etu kehittäjän kannalta on, että Android-kehittäjän ei tarvitse lisätä sovellukseensa koodia, joka käsittelee maksutapahtumia. Kehittäjä ei tarvitse esimerkiksi maksutapahtumien tietoturvaan liittyvää kokemusta, sillä maksuliikenne tapahtuu aina Google Play -sovelluksen kautta. Käyttäjän kannalta ostamiskokemus on aina samanlainen jokaisessa Android-sovelluksessa. (12.)

Google Play tukee kahdenlaisia sovelluksensisäisiä ostoksia: hallinnoituja ostoksia (managed in-app products) sekä tilauksia (subscriptions). (13.)

### Hallinnoidut sovelluksen sisäiset ostokset

Hallinnoidut ostokset tallentuvat Google Playlle ostohetkellä. Sovelluksen sisältä voidaan lähettää kyselyjä Google Playlle, mitä kaikkea käyttäjä on ostanut. Esimerkiksi sovellusta käynnistettäessä voidaan kysellä, onko käyttäjä ostanut premium-version sovelluksesta. Jos on, sovelluksen käyttöliittymää voidaan muuttaa premium-sovelluksen käyttäjälle asianmukaisesti.

Google Playlle voi myös lähettää kyselyjä tuotteen kuluttamisesta. Sovelluskehittäjä voi haluta tehdä kulutettavia tuotteita esimerkiksi videosovelluksessa, jossa käyttäjät voivat ostaa videoiden tai elokuvien katselukertoja.

Aina kun käyttäjä ostaa tuotteen, Google Play asettaa ostetun tuotteen omistettu-tilaan. Omistettu-tilassa olevaa tuotetta ei voi ostaa uudelleen. Google Playlle voi lähettää kyselyn tuotteen kuluttamisesta, jonka jälkeen tuote asetetaan ei-omistettu-tilaan, onka jälkeen tuotteen voi ostaa uudestaan.

### Ostotapahtuman eteneminen

Ostotapahtuman tietoliikenne kuvaillaan kuvassa 2. Kehittäjän tekemän sovellus on kuvan vasemmalla puolella, oikealla puolella on Google Play -sovellus. Aika alkaa ylhäältä ja etenee alaspäin. isBillingSupported-metodilla kysytään, onko sovelluksen käyttämä sovelluksen sisäisten ostosten rajapinta Google Playn tukema.

getPurchases-metodilla kysytään mitä tuotteita käyttäjä omistaa. Vastaus palauttaa paketin jossa on listattu käyttäjän omistamat tuotteet.

getSkuDetails-metodilla haetaan lista kaikista tuotteista jotka on mahdollista ostaa. ”Sku” tarkoittaa Stock Keeping Unit, eli varastointinimike, mikä tarkoittaa tuotteen ainutlaatuista tunnistetta. Kehittäjä voi määritellä tuotteet kuvauksineen ja hintoineen Google Play -kehittäjäkonsolilla.

getBuyIntent-metodille annetaan ostettavan tuotteen tunniste ID, jolla itse ostotapahtuma aloitetaan. Google Play palauttaa PendingIntent-tyyppisen olion, joka luo sovelluksen käyttöliittymässä Google Play-kaupan ponnahdusikkunan, jossa osto voidaan hyväksyä.

Kun ostotapahtuma on päättynyt ja Google Playn ostoaktiviteetti päättyy, Google Play lähettää Intent-tyyppisen olion aktiviteetin onActivityResult-metodiin. onActivityResult-metodissa on resultCode-kokonaislukumuuttuja, jonka arvo tiedottaa ostotapahtuman onnistumisesta. Kehittäjä voi lukea tiedot jotka ovat onActivityResult-metodissa, ja tehdä sovelluksessa muutokset joilla käyttäjä saa ostamansa tuotteen käyttöön.



**Kuva 2.** Ostotapahtuman tietoliikenne. (13.)

### Tilaukset

Samaan tapaan kuin sovelluksen sisäiset tuotteet, tilaukset määritellään Google Play kehittäjäkonsolissa. Kehittäjä voi määritellä tilauksen hinnan, sekä tilauksen rahastuksen tiheyden. (9.)

Ostotapahtuman eteneminen toimii samaan tapaan tilausta tehtäessä kuin hallinnoituja sovelluksen sisäisiä ostoksia tehtäessä. Ainoa ero on, että tilausta tehtäessä koodissa on annettava getBuyIntent-metodille tuotteen tyyppi -parametriksi merkkijono ”subs” (13):

**Bundle bundle = mService.getBuyIntent(3, "com.example.myapp",  
   MY\_SKU, "subs", developerPayload);**

Sovelluksessa voidaan kysellä aktiivisia tilauksia getPurchases-metodilla (13):

**Bundle activeSubs = mService.getPurchases(3, "com.example.myapp",  
                   "subs", continueToken);**

## MAINOKSET

### Käyttöönotto

Admob-mainokset on helppo lisätä omaan Android-sovellukseen. Build.gradle-tiedoston dependencies-osioon on lisättävä seuraava (14):

**compile 'com.google.android.gms:play-services-ads:8.4.0'**

Tällöin Android Studio lisää projektiin Google Admob -kirjastot.

Kehittäjä voi vaikuttaa mainosten sisältöön Google Play -kehittäjäkonsolilla. Halutessaan kehittäjä voi suodattaa pois sovelluksensa mainonnasta tiettyihin aihealueisiin kuuluvat mainokset. Konsolilla voi myös nähdä tilastoja mainosten kannattavuudesta.

### Mainosbannerit

Sovelluksen käyttöliittymän määritteleviin xml-tiedostoihin voidaan lisätä adView-näkymä, joka näyttää mainosbannerin sisällön käyttäjälle. Mainoksiin voidaan hakea sisältö aktiviteetin onCreate-metodissa (14).

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 **AdView mAdView = (AdView) findViewById(R.id.adView);**  
 **AdRequest adRequest = new AdRequest.Builder().build();**  
 **mAdView.loadAd(adRequest);**

}

### Koko näytön mainokset

Admob tukee myös koko näytön mainoksia (interstitial ads), joita voidaan näyttää aina kun käyttäjä haluaa siirtyä sovelluksessa näkymästä toiseen. Kun mainos suljetaan, käyttäjä pääsee seuraavaan näkymään. (15.)

Java-koodissa InterstitialAd-luokassa on setAdListener-metodi, jolla voidaan määritellä, mitä tapahtuu, kun käyttäjä sulkee mainoksen (esimerkiksi toiseen näkymään siirtyminen). requestNewInterstitial-metodilla palvelu hakee uuden kokonäytön mainoksen. Metodia kannattaa kutsua heti mainoksen sulkeutuessa, sillä järjestelmä kykenee lataamaan taustalla uutta mainosta asynkronisesti. Jos mainosta alettaisiin hakemaan, kun uusi mainos ilmestyy ruutuun, käyttäjän täytyisi odottaa sen latautumista. (15.)

# JOHTOPÄÄTÖKSET

Tämän työn tarkoitus oli arvioida mobiilisovellusmarkkinoita liiketoiminnan kannalta. Kuten liiketoiminnassa aina, liiketoimintasuunnitelma on tehtävä hyvin huolellisesti riskianalyyseineen, ja on harkittava tarkkaan tilastojen avulla kannattaako yritystoimintaan lähteä.

Vaikka Android-sovellusten kehittäminen on helppoa ja nopeaa, sovelluksen tekeminen yksinään on silti työlästä, ja voi kestää kuukausia tai jopa vuosia. Kuukausittaisen toimeentulon tienaaminen Android-sovelluksella ei riitä, sillä kehittäjän täytyy kyetä tienaamaan vuosia kestäneen sovelluskehityksen kulut takaisin. Liiketoimintasuunnitelman ja riskianalyysin laiminlyöminen johtaa kaikin puolin kannattamattomaan yritystoimintaan.

Mobiilisovelluskehittäminen on aina suuri riski. 64 % Android-sovellusten kehittäjistä tienaavat vähemmän kuin 500$ jokaista julkaisemaansa sovellustaan kohden kuukaudessa (15). Yritystoimintaa suunnittelevalle luku voi vaikuttaa lannistavalta.

Tuotetta kehittäessä, sekä yritystoimintaa aloittaessa, yrittäjän on aina muistettava, että riski on osa yritystoimintaa. Jos riskejä ei uskalla ottaa, tuotteesta tulee liian tavanomainen, eivätkä asiakkaat ole kiinnostuneita siitä. Liiketoimintaa voi olla mahdoton ennustaa, ja mahdollisuus menestymisestä on aina olemassa. Jopa tuotteesta, jonka pitäisi epäonnistua kaikkien laskelmien ja ennusteiden mukaan, voi tulla suuri menestys.

# LÄHTEET

1. van der Meulen, Rob — Rivera, Janessa 2015. Gartner Says Worldwide Smartphone Sales Recorded Slowest Growth Rate Since. Gartner, Inc. Saatavissa: <http://www.gartner.com/newsroom/id/3115517> Hakupäivä: 18.4.2016
2. Earn. Android Developers. Saatavissa: <http://developer.android.com/distribute/monetize/index.html> Hakupäivä: 5.2.2016
3. Vannieuwenborg, Frederic — Mainil, Laurent — Verbrugge, Sofie — Pickavet Mario — Colle, Didier 2012. Business models for the mobile application market from a developer’s viewpoint. 16th International Conference on Intelligence in Next Generation Networks. Saatavissa: IEEE Explore -tietokanta (vaatii käyttöoikeuden). Hakupäivä 12.2.2016.
4. Adding AdMob into an Existing App. Google Developers. 2016. Saatavissa: <https://developers.google.com/admob/android/existing-app> Hakupäivä: 16.4.2016
5. Chaffey, Dave 2006 — Internet Marketing: Strategy, Implementation and Practice. Harlow: Pearson Education Limited. Saatavissa: <https://books.google.fi/books?id=G9smMWZ-DWgC> (rajoitettu lukuoikeus). Hakupäivä: 16.4.2016
6. Why AdMob?, Admob By Google, Saatavissa: <https://www.google.com/admob/monetize.html> Hakupäivä: 16.4.2016
7. Android platform/API version distribution, Android Studio 1.5. Saatavissa: <http://developer.android.com/sdk/index.html> Hakupäivä: 19.3.2016.
8. Launch Checklist. Android Developers. Saatavissa: <http://developer.android.com/distribute/tools/launch-checklist.html> Hakupäivä: 19.3.2016.
9. Developer program FAQ, Microsoft Developer resources, Saatavissa: <https://dev.windows.com/en-us/programs/faq> Hakupäivä 16.4.2016
10. Program Membership Details, Apple Developer Program. Saatavissa: <https://developer.apple.com/programs/whats-included> Hakupäivä 16.4.2016
11. Publishing Overview. Android developers. Saatavissa: <http://developer.android.com/tools/publishing/publishing_overview.html>
12. In-app Billing. Android developers. Saatavissa: <http://developer.android.com/google/play/billing/index.html> Hakupäivä 16.4.2016
13. In-app Billing API. Android developers. Saatavissa: <http://developer.android.com/google/play/billing/api.html> Hakupäivä 16.4.2016
14. Get Started in Android Studio. AdMob for Android. Saatavissa: <https://developers.google.com/admob/android/quick-start> Hakupäivä: 16.4.2016
15. Interstitial Ads. AdMob for Android. Saatavissa: <https://developers.google.com/admob/android/interstitial> Hakupäivä: 16.4.2016
16. Wilcox, Mark 1014. State of the Developer Nation: The App Economy Consolidates Before the Next Gold Rush. Saatavissa: <http://www.developereconomics.com/the-app-economy-consolidates-before-the-next-gold-rush/> Hakupäivä 16.4.2016