Lauri Miettinen

**KLASSISTEN AUTOJEN ELINKAAREN SEURANTA ÄLYKKÄÄLLÄ SOPIMUKSELLA**

**KLASSISTEN AUTOJEN ELINKAAREN SEURANTA ÄLYKKÄÄLLÄ SOPIMUKSELLA**

Lauri Miettinen

Opinnäytetyö

Syksy 2016

Kone- ja tuotantotekniikan koulutusohjelma

Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu

Koulutusohjelma, suuntautumisvaihtoehto

Tekijä(t):

Opinnäytetyön nimi:

Työn ohjaaja(t):

Työn valmistumislukukausi ja -vuosi: Sivumäärä:60 + 5 liitettä

Tiivistelmä on suppea, itsenäinen esitys, joka antaa lukijalle kuvan tehdystä työstä. Se kirjoitetaan sitten, kun työ on valmis. Tiivistelmä kirjoitetaan täydellisin virkkein ja passiivimuodossa. Siinä selostetaan lyhyesti työn aihe, tavoitteet, työn keskeisin toteutustapa ja olennaiset tulokset.

Sisältö jakaantuu yleensä kolmeen kappaleeseen:

1. työn aihe ja tavoitteet

2. menetelmät, toteutustapa ja eteneminen

3. tulokset ja esitetyt johtopäätökset.

Tiivistelmän tulee mahtua yhdelle sivulle rivivälillä 1.

Korvaa nämä tekstit omillasi.

Asiasanat: x, x, x

(Kirjoita 3–7 kpl työtäsi kuvaavaa asiasanaa*.* Käytä hyväksesi esimerkiksi yleistä suomalaista asiasanastoa YSA, <http://onki.fi/fi/browser/overview/ysa>.)

ABSTRACT

Oulu University of Applied Sciences

Degree programme, option

Author(s):

Title of thesis:

Supervisor(s):

Term and year when the thesis was submitted: Pages: 60 + 5 appendices

Vieraskielinen tiivistelmä (Abstract) tehdään työn tilaajan niin vaatiessa tai osaston käytänteiden mukaisesti. Sen ei tarvitse olla suora käännös suomenkielisestä tiivistelmästä, vaan sen voi kirjoittaa kansainväliselle lukijalle.

Remember to include the following in the abstract:

Why – background for the thesis, assigner

What – objective of the thesis

How – background knowledge applied, research methods, material used

Achievements – main results

Exploitability – conclusions, development proposals, further measures

The length of the abstract is 1 page.

Keywords: x, x, x

(List 3–7 keywords that describe your thesis. Use for example keywords from the following controlled vocabulary thesauri:

MeSH <http://www.yso.fi/onto/mesh/conceptscheme>

Agriforest <http://www-db.helsinki.fi/agri/agrisanasto/Welcome_eng.html>

Helecon <http://helecon3.hkkk.fi/helevoc/?lang=eng&dbname=MIX>)

ALKULAUSE

Raportissa voi olla alkusanat, joissa kiitetään toimeksiantajaa, ohjaajia ja mahdollisesti muita tukijoita tai avustajia. Alkusanat voi sisältää lyhyesti myös muuta työhön liittyvää, lukijaa kiinnostavaa tietoa, joka ei muuten tule raportissa esiin. Alkusanat päivätään ja allekirjoitetaan.

Korvaa nämä tekstit omillasi.

SISÄLLYS

Contents

[1 Johdanto 8](#_Toc480806699)

[2 SUUNNITTELU JA KÄYTTÖTAPAUKSET 11](#_Toc480806700)

[2.1 Lohkoketjun hyöty sovelluksessa 11](#_Toc480806701)

[3 OHJELMISTOON TUTUSTUMINEN 13](#_Toc480806702)

[3.1 GoEthereum (geth) 13](#_Toc480806703)

[3.2 Ethereum Wallet 13](#_Toc480806704)

[3.3 Browser Solidity 13](#_Toc480806705)

[3.4 Truffle-sovelluskehys 13](#_Toc480806706)

[3.5 Meteor 14](#_Toc480806707)

[4 TYÖN KULKU 15](#_Toc480806708)

[5 LOPPUTULOKSET 16](#_Toc480806709)

[5.1 Toteutumattomia käyttötapauksia 16](#_Toc480806710)

[6 yhteenveto 17](#_Toc480806711)

SANASTO

Jos työssä toistuvat samat lyhenteet tai suureiden, yksiköiden tai piirrosten merkinnät ja jos niitä on paljon, niistä kootaan luettelo selityksineen. Otsikkona voi olla sisällön mukaan esimerkiksi LYHENTEET, MERKKIEN SELITYKSET tai SANASTO. Luettelo laaditaan aakkosjärjestykseen. Aakkostuksessa noudatetaan SFS-standardia 4600 ja SI-järjestelmää.

Posta tämä sivu, jos et tarvitse.

# Johdanto

Omat kommenttini vihreällä

Työn taustassa on tärkeää selittää seuraavat asiat:

1. Mikä on lohkoketju? Mistä se sai alkunsa? Tämä ei ole kovin selvä asia edes monille teknisille ihmisille, joten paras selittää hyvin.
2. Lohkoketjut on helpompi ymmärtää jos selittää Bitcoinin taustan ja historian. Selitä ne lyhyesti.
3. Selitä Ethereum. Mikä se on, mikä sen idea on ja mistä se sai alkunsa?
4. Selitä itse työn tausta-ajatus. Mikä idea on tehdä lohkoketjuihin liittyvä sovellus vintage-autoista?
5. Kuvaile Klassisten autojen kulttuuria. Mitä piirteitä sillä on, ja mitä nämä ihmiset arvostavat?
6. Pohdiskele klassisten autojen kulttuurin arvoja, ja vakuuta lukija siitä, että lohkoketju on hyvä lisä osaksi kulttuuria.
7. Miksi juuri lohkoketjut? Miksei tätä järjestelmää voi tehdä perinteisellä tietokannalla? Pohdi, miten klassisten autojen kulttuuri hyötyy lohkoketjuista.

Eli tekstiä tulee paljon. Pitänee jakaa johdantokappale useampaan osaan. Ehkä voisi kirjoittaa johdantoon lyhyesti tuo neloskohta, ja sitten myöhemmissä kappaleissa lohkoketjujen historia?

Lohkoketjutekniikka sai alkunsa vuonna 2009, kun Satoshi Nakamoto kirjoitti paperin, jossa hän ehdotti tietomallin puolueettoman valuutan luomisesta. Nykyään valuutta liikkuu pankin tai muun laitoksen kautta. Bitcoinin idea oli luoda valuutta josta ei ole vastuussa kukaan keskitetty taho, jossa maksajan ei tarvitse luottaa keneenkään. Nakamoto suunnitteli hajautetun järjestelmän rahan lähettämiselle ja maksutapahtumien todentamiselle. (1)

Bitcoin- maksut tallennetaan hajautettuun tietokantaan, jota kutsutaan lohkoketjuksi. Kukaan yksi yritys tai yksi taho ei ole vastuussa Bitcoin-maksuista. Kukin maksutapahtuma lähetetään vertaisverkkoon, jossa se todennetaan. Kuka tahansa voi liittää tietokoneensa Bitcoin-verkkoon todentamaan maksutapahtumia. Maksutapahtumat todennetaan menetelmällä, joka vaatii paljon laskentatehoa. Järjestelmän huijaaminen vaatisi hyökkääjältä enemmän laskentatehoa kuin mitä on kaikkilla verkossa olevilla rehellisillä osallisilla. Bitcoin-verkkoon hyökkääjän on ainakin teoriassa mahdollista luoda itselleen rahaa, jota hänellä ei oikeasti ole. Käytännössä hyökkääjällä pitäisi silloin olla käytössään enemmän laskentatehoa kuin koko muulla maailmalla. (1.)

Lohkoketju on arvokkaan tiedon tallentamiseen perustuva järjestelmä. Bitcoinin tapauksessa arvokas tieto on varallisuus ja valuutta.

3)

Bitcoinin tekniikka herätti maailmalla mielenkiintoa. Nakamoto pohti hajautettua valuuttaa suunnitellessaan, että olisi suunnitellut laajan hajautetun ohjelmointikielen lohkoketjuille, mutta päättikin tehdä yksinkertaisemman järjestelmän, sillä ymmärsi tekniikan olevan kokeellinen ja haastava luonteeltaan. (2.)

Siitä Vitalik Buterin keksi tehdä Ethereumin, ohjelmointikielen ja sovellusalustan, jonka sovellukset pyörivät lohkoketjussa (3.) Kenties joku parempi lähde tähän. Ehkä se alkuperäinen Buterinin paperi?

Klassiset autot ovat arvokkaita. Niistä voidaan käydä satojen tuhansien (miljoonien?) eurojen kauppoja. Autoharrastajat kokevat, että autot ovat maansa kulttuurille merkityksellisiä, sillä kussakin klassisessa autossa on viitteitä aikansa kulttuurista.

Klassiset autot liittyvät myös liikennöinnin alaan. Liikennöinnin alallakin on paljon tulevaisuuden sovelluksia, joissa voitaisiin hyödyntää lohkoketjuja.

Autoja huoltaessa huoltotietojen tallennus olisi haviteltava ominaisuus. Sekä tavallisille autoille, että klassisille autoille. Tavallisilla autoilla huoltotietoja voivat hyödyntää vakuutusyhtiöt, sekä auton omistaja, jotta pysyisi ajan tasalla auton kunnosta ja muistaisi huoltotoimenpiteet.

Klassisia autoja on huollettava, sillä ne voivat olla kymmeniä vuosia vanhoja, ja osat kuluvat väistämättä ajaessa. Lohkoketjuun tallennettua tietoa on vaikea muokata jälkeenpäin. Jos klassisen auton huoltotiedot, käyttötilastot (kilometrimittari) tallennettaisiin lohkoketjuun, auton omistaja voisi vakuuttaa huutokaupoissa ostajan autonsa arvosta.

Nykyään älykkäät sopimukset eivät vielä ole lyöneet tulille (paranna). Tässä työssä oli tarkoitus tehdä käytännön sovellus. Työn perimmäinen tarkoitus oli luoda konkreettinen kuva siitä, mihin älykkäitä sopimuksia voitaisiin käyttää tulevaisuudessa.

Työssä pohditaan liiketoimintamalli ja käyttötapaukset lohkoketjutekniikkaa hyödyntävälle sovellukselle. Tavoite oli luoda Ethereum-lohkoketjuun älykäs sopimus, jota voitaisiin hyödyntää klassisen auton elinkaaren seurantaan. Toinen osa työtä oli luoda web-käyttöliittymä jolla lohkoketjuun tehtäviä merkintöjä voi tehdä ja tarkastella. Tässä työssä raportoidaan työn suunnittelusta, totetuksesta ja lopputuloksesta.

# SUUNNITTELU JA KÄYTTÖTAPAUKSET

Ajatuksena oli tehdä yleisiä merkintöjä. Merkinnöistä käytetään nimeä kohokohta, eli highlight.

Auton omistajalla on käytössään auton elinkaarta seuraava älykäs sopimus.

Ulkopuoliset käyttäjät voivat tehdä verkkosovelluksella kohokohtapyyntöjä (highlight request) älykkääseen sopimukseen. Kohokohtapyyntönä voi olla esimerkiksi auton huoltaminen korjaamolla tai asiantuntija-arvio auton autenttisuudesta alkuperälleen. Kohokohtapyynnön tekijä voi pyytää pienen rahallisen korvauksen vaivannäöstä.

Merkintöjen tavoite on vakuuttaa tuleva ostaja auton arvokkuudesta. Vaikka kohokohtapyynnöt maksaisivat auton omistajalle useita euroja tai kymmeniäkin euroja per merkintä, ne maksavat itsensä takaisin moninkertaisesti, mikäli auton ostaja päättää tarjota huutokaupassa suuren summan autosta, johtuen arvosta jonka merkinnät tuovat autolle.

## Lohkoketjun hyöty sovelluksessa

Mitä hyötyä on juuri lohkoketjusovelluksella? Miksei tätä voisi tehdä ihan perinteisenä sovelluksena, jossa merkinnät tallennetaan tietokantaan?

Yksi etu lohkoketjussa on tietysti se, että lohkoketjusovelluksessa ei ole lainkaan palvelinkustannuksia. Tavallisessa palvelussa palvelinkustannukset täytyy tienata takaisin liiketoiminnalle, kuten mainoksilla. Tämä palvelu on luonteeltaan sellainen, että sitä ei välttämättä käytetä kovin usein. Liiketoiminnan tekeminen voi olla haastavaa

Toinen etu on lohkoketjumerkintöjen pitkäikäisyys. Klassisen auton elinkaari voi olla vuosikymmeniä, eivätkä kovin monet yritykset ole olemassa niin kauaa. Jos tavallinen palvelu menee nurin, tietokannassa olevat merkinnät katoaisivat sen mukana. Lohkoketju on paljon pitkäikäisempi, ja koska kopiot tiedoista ovat läsnä jokaisella ketjua käyttävällä tietokoneella, tiedot eivät edes voi tuhoutua lopullisesti.

Lohkoketjussa kaikki maksutapahtumat ovat julkisia ja läpinäkyviä. Läpinäkyvyys sopii tämän sovelluksen käyttötapauksiin hyvin. Jos kaikki auton elinkaaressa tulleet tapahtumat ovat julkisia, on huijausten ja väärinkäytösten tekeminen kannattamatonta auton omistajalle. Sovelluksella tehtyjä merkintöjä tarkkailevat voivat tehdä tutkimusta, ja havaita jos auton elinkaaren merkinnöissä esiintyy väärää tietoa tai ristiriitoja. Sosiaalisen median aikakautena tieto petollisesta auton omistajasta leviää nopeasti ihmiseltä ihmiselle. Maineen menetyksen pelko voi olla suuri. Mikäli tällainen sovellus on oikeasti käytössä tulevaisuudessa, autonomistajat varmasti ymmärtävät, että on kannattavaa pysyä rehellisenä.

# OHJELMISTOON TUTUSTUMINEN

## GoEthereum (geth)

Paikallinen geth-solmu oli painajaismaisen hidas. Kuulin testrpc-ohjelmasta (onko parasta kutsua sitä ”ohjelmaksi”? Palvelin? Sovellus?).

## Ethereum Wallet

## Browser Solidity

## Truffle-sovelluskehys

Truffle oli hanurista.

Tehtiin alkuvaiheessa automaattisia testejä. Automaattisten testien koodaus oli kumminkin hyvin työlästä, joten niiden kehittämisestä luovuttiin myöhemmin.

## Meteor

Meteor. Ensin harjoiteltiin Meteorin käyttöä tekemällä lohkoketjuihin liittymätön harjoitussovellus. Tämä työ käytti MongoDB-tietokantaa.

# TYÖN KULKU

Tässä vaiheessa voisin ehkä kertoa, mitä ohjelmistoja päädyin käyttämään.

Esittele kuvakaappauksia

# LOPPUTULOKSET

Kuvaile tänne mitä tuli tehtyä, miten toimii ja miten onnistui.

Laita kuvakaappaus valmiista työstä.

## Toteutuneet käyttötapaukset

Ehkä tähän voisi lyhyesti listata käyttötapaukset? Esitellään ne käyttötapaukset jossain muualla.

## Toteutumattomia käyttötapauksia

- Spämmäyksen estojärjestelmä. Highlight request rights. Voi antaa oikeuksia tietyille tilille tehdä highlighteja. Ainoastaan ulkopuoliset käyttäjät, joilla on oikeus, voivat lähettää highlight requesteja. Strict highlight request mode. Voi ottaa pois päältä jos auto on epäsuosittu. Jos auto saa näkyvyyttä enemmän lehdistössä tai internetissä, strict moden voi laittaa päälle.

Yhdessä ohjelman versiossa tätä kehitettiin, mutta kehitys lopetettiin, koska koettiin, ettei käyttötapauksen demonstroiminen ollut tarpeeksi tärkeää työn tarkoituksen kannalta.

- Eventien filtteröinti ja järjestely. Lopullisessa sovelluksessa kaikki ovat samassa listassa. Olisi voinut käyttää local collectioneitä, niin olisi päässyt filtteröimään. Tässä työssä on yhdessä esittelyssä melko vähän tavaraa, niin ei ehkä niin haittaa, ettei tullut tehtyä tätä.

* Uusien ajoneuvojen lisääminen sovelluksella. Näytä kuva siitä puolivalmiista käyttöliittymästä.

# yhteenveto

Korvaa teksti jälleen omilla teksteilläsi.

Työn viimeisessä numeroidussa luvussa toistetaan työn päätarkoitus, tulokset ja saatu uusi tieto sekä pohditaan sen ongelmia, tuloksia ja päätelmiä. Tuloksia verrataan lähtötietomuistiossa ja johdannossa asetettuihin tavoitteisiin. Tuloksiin vaikuttaneita seikkoja pohditaan kriittisesti. Luvun otsikoksi voidaan antaa esimerkiksi YHTEENVETO, POHDINTA tai LOPPUSANAT.

Tässä työssä Truffle-sovelluskehyksestä saatu hyöty jäi vähäiseksi.

Solidity-kieli jätti toivomisen varaa. Kielessä on paljon teknisiä rajoitteita. Osa rajoitteista johtuu ohjelmointikielen vähäisestä kehityksestä. Solidity on tällä hetkellä versiossa 0.4.7. Osa solidity-kielen rajoitteista johtuu Ethereum-virtuaalikoneen rajoitteista.

Ei voi kutsua vieraassa sopimuksessa olevaa funktiota, joka palauttaa string-tyyppisen muuttujan. Kerro mistä johtuu ja miten tämä rajoitti työskentelyä.

Ei voi hakea koko mapping-tietorakenteen kaikkia alustettuja alkioita. Kehittäjän täytyy itse kirjoittaa funktiot koko mappingin alkoiden haulle. Olisi mahdollista, että kuka tahansa ihminen maailmassa voisi tehdä kotisivun/sovelluksen, jokatoimisi ikään kuin käyttöliittymänä Classic Car Chain -sovellukselle. Mutta sovelluksessa on varta vasten tehty koodia koko mapping-tietorakenteen haulle, joka voi olla vaikea kopioida. Tämä tulee hidastamaan huomattavasti hajautettujen sovellusten kehitystä, niin kauan kuin se on olemassa.

Pieneltä kuulostavat rajoitteet kumminkin rajoittavat suuresti kaikkia mahdollisia käyttötapauksia, mitä Ethereum-alustalla voisi muuten toteuttaa. Ellei rajoitteita tulla korjaamaan, voi olla, ettei Ethereum koskaan tee läpimurtoa mitä sen perustajat haluavat sen tekevän. Tästä huolimatta, Soldity-kieltä kehitetään parhaillaan, ja joitakin hyvin oleellisia parannuksia on tehty viime aikoinakin, kuten (etsi se joku foorumi missä ihmiset toivoivat jotain ominaisuutta ja linkkaa dokumentaatiossa johonkin paikkaan, missä todistat, että ominaisuus on jo tehty).

Silti kukaan ei voi kieltää tekniikan mahdollisuuksia. Koska kieli on niin yksinkertainen ja kuka tahansa jolla on ymmärrystä ohjelmoinnista voi oppia sen. Kielen helppokäyttöisyydellä ja yksinkertaisuudella voi olla suuri merkitys Ethereum-alustan maailmanlaajuisessa läpimurrossa.

LÄHteet

1. <https://bitcoin.org/bitcoin.pdf>
2. <https://www.youtube.com/watch?v=3PdO7zVqOwc>
3. <http://futurethinkers.org/vitalik-buterin-ethereum-decentralized-future/>

LIITTEET

Liite 1 Lähtötietomuistio

**Tekijä** Lauri Miettinen

**Tilaaja** Oulun Ammattikorkeakoulu, tekniikan laitos

**Tilaajan yhteyshenkilö(t) ja yhteystiedot**

Janne Kumpuoja ([Janne.Kumpuoja@oamk.fi](mailto:Janne.Kumpuoja@oamk.fi))**,**

Veijo Väisänen (Veijo.Vaisanen@oamk.fi)

**Projektin nimi**

Vintage-auton elinkaaren seuranta lohkoketjutekniikalla -opinnäytetyö

**Projektin tavoitteet**

Tavoite on luoda Ethereum-lohkoketjukehitysalustan avulla älykäs sopimus, jota voitaisiin hyödyntää Vintage-auton elinkaaren seurantaan. Tavoitteena on tehdä web-käyttöliittymä, jonka avulla käyttäjä voi vuorovaikuttaa lohkoketjun kanssa. Työn tuloksista raportoidaan opinnäytetyödokumentissa.

Työssä suunnitellaan ja esitellään liiketoimintamalli, jota valmis tuote voisi käyttää.

Opinnäytetyödokumentissa kuvaillaan luotua järjestelmää ja esitellään sen taustalla oleva tekniikka. Lukijalle esitellään lyhyesti lohkoketjutekniikan perusteet ja historia.

**Projektissa käytettävä prosessimalli (mahdollinen vaihejakoon perustuva aikataulutus)**

28.9.2016: Harjoitustyön vaatimusmäärittelydokumentti oltava valmiina.

21.10.2016: Sprint 1 release. Projektin pohjimmaiset järjestelmät, sovelluskehykset ja palvelimet oltava käyttö- ja kehityskelpoisia.

16.12.2016: Sprint 2 release. Sovelluksesta oltava valmiina versio, joka on esittelykelpoinen (minimum effort). Kirjallisessa työssä oltava kuvauksia projektin etenemisestä.

10.3.2017: Sprint 3 release. Sovelluksen minimituote (minimum viable product) oltava esittelykepoinen.

28.4.2017: Sprint 4 release. Kirjallinen työ, sovellus ja liiketoimintamalli on oltava valmiina. Pidetään lopetuspalaveri.

**Projektissa käytettävät menetelmät ja teknologiat**

Ethereum; scrum-menetelmä; Github –versionhallinta; Meteor, tai muu Javascript-pohjainen sovelluskehys.

**Projektin alustava aikataulu** Työ alkaa 13.9.2016, ja päättyy 30.4.2017