Miten sähköistä ylioppilaskoetta voisi kehittää?

Roni Juntunen

Tammikuu 2017

Sisällys

Lisensointi

1. Johdanto

Digitaalisen ylioppilasjärjestelmän perusta on jo hyvässä vauhdissa ja testattu kerran todellisessa tilanteessa. Järjestelmä ei selvästikään ole vielä valmis, mutta siitä löytyy jo paljon hyvää. Esimerkiksi kokelaiden kohtaama selaimeen aukeava sivusto on selkeäkäyttöinen ja siistin näköinen. Toisaalta myös edelleen suuria ongelmia löytyy muun muassa ohjelmiston ja laitteistojen yhteensopivuuden kanssa ja ylipäätään käyttäjäystävällisyydessä varsinkin käynnistyksen yhteydessä.

Varsinkin yhteensopivuusongelmat tulisi ratkaista ensi tilassa, sillä ne hankaloittavat tai jopa estävät joidenkin koneiden käyttöä. Samalla tulisi parantaa myös käyttäjäystävällisyyttä erityisesti käynnistyksen yhteydessä, jotta Abitti-koejärjestelmä olisi mahdollisimman käytännöllinen kaikenlaisissa koetilanteissa.

1.1. Tutkimusongelma

Ongelmana on siis saada jo olemassa olevat, koejärjestelmän hyvät ominaisuudet uuteen pakettiin. Paketin tulisi siis olla yksinkertainen, nopea ja turvallinen. Tähän tilanteeseen pääsemiseksi on siis jotenkin päästävä eroon nykyisestä käynnistämismetodista eli USB-muistitikuista.

2. Erilaisia metodeja

2.1. Selite

Tässä osiossa pohdin erilaisia tapoja, joilla Abitti-järjestelmän käynnistyksen voisi hoitaa jotenkin toisin. Kaikki ideat perustuvat siihen, että nykyistä järjestelmää voisi käyttää vielä mahdollisimman laajalti hyödyksi.

2.2. PXE

2.2.1. Info

PXE (Preboot eXecution Environment), tarkoittaa käynnistystä edeltävää ajoympäristöä. PXE:tä käytetään useimmiten verkon yli tapahtuvaan tietokoneen käynnistykseen. PXE löytyy melkein kaikista konemalleista, koska se on jo verrattain vanha standardi. Lisäksi PXE on 2015 vuodesta lähtien osa UEFI (Unified Extensible Firmware Interface) standardia, joten PXE:n tulisi sisältyä teoriassa siis kaikkiin uusiin konemalleihin.

2.2.2. Toteutus

PXE vaatii toimiakseen siis langallisen verkon, jossa on yksi PXE palveluita tarjoava palvelinkone ja erilaisia käynnistettäviä asiakaskoneita, eli kokelaiden tietokoneita. PXE-serverikokonaisuus itsessään muodostuu kahdesta erilaisesta serveriohjelmasta, jotka ovat DHCP- ja TFTP-palvelin. Molemmista näistä ohjelmista löytyy täysin avoimet variaatiot, kuten dhcp3 ja atftpd. Palvelimen muokkausten lisäksi kokelaiden koneiden UEFI tai BIOS (Basic Input Output Settings) asetuksista tulisi sallia PXE-käynnistys ja laittaa se käynnistysjärjestyksessä ensimmäiseksi.

2.2.3. Hyvät puolet

Tämän käynnistystavan ehdoton hyvä puoli on se, että se poistaisi muistitikkujen tarpeen ja olisi melko helppo lisätä käynnistysvaihtoehdoksi jo olemassa olevaan järjestelmään. Lisäksi turvallisuus tulisi olemaan todennäköisesti huomattavasti parempi tikkujen eliminoimisen jälkeen, koska silloin koko järjestelmän joutuisi lataamaan hyvin alhaisella tasolla suoraan palvelinkoneelta, eikä esimerkiksi nopea tikkujen väärentäminen olisi mahdollista. Myös valekäyttöjärjestelmän ajaminen virtuaalikoneen avustuksella voitaisiin eliminoida, mikäli esimerkiksi palvelinkone tutkisi mac-osoitteiden perusteella, kuinka moni kone on käyttöjärjestelmän jo palvelinkoneelta ladannut.

2.2.4. Huonot puolet

PXE-käynnistystä käyttämällä menetettäisiin mahdollisuus tikulle tietojen tallentamiseen. Haittapuolen voisi välttää esimerkiksi siten, että kokelaita pyydettäisiin tuomaan tyhjä muistitikku, joka valittaisiin käyttöön ja formatoitaisiin Digabi-käyttöjärjestelmän käynnistyttyä jonkinlaisen valikon avustuksella. PXE-käynnistys kuormittaisi myös huomattavasti enemmän langallista verkkoa ja palvelinkoneen massamuistia kokeen alussa. Palvelinkoneen levynkulutus olisi luultavasti sitä tasoa, että palvelinta ei voisi pitää yllä muistitikulta, vaan palvelimen käyttöjärjestelmä pitäisi paremminkin olla tietokoneen sisäisellä ssd-levyllä.

2.2.5. Käyttäjäkunta

Tämä uudistus hyödyttäisi hieman itse ylioppilaskirjoituksia, kun tikkuja ei tarvitsisi luoda ainakaan yhtä paljon kuin ennen. Uudistus voisi olla myös erittäin kätevä vaikkapa kurssikoetta suoritettaessa, mikäli saatavilla olisi langallinen verkko. Silloin parhaassa tapauksessa pelkästään palvelintikkua tai sisäistä kovalevyä tarvitsisi päivittää, mikä nopeuttaisi ja helpottaisi kokeiden järjestämistä. Tosin useimmat nopeat kokeet järjestetään ihan tavallisessa luokassa wifi-verkon avustuksella, jolloin PXE-käynnistystä ei vain yksinkertaisesti voisi hyödyntää.

2.2.6. Kustannukset

Järjestelmän toteuttaminen ei aiheuta merkittäviä kustannuksia koejärjestelmän laatijoille. Se aiheuttaa kuitenkin jonkin verran kustannuksia kokeita ylläpitäville kunnille palvelinkoneen levykapasiteetin vaatimusten takia.

2.2.7. Lopputulema

PXE-käynnistyksellä voitaisiin kohtuullisen helposti lisätä turvallisuutta itse ylioppilaskirjoituksissa, mutta muita ongelmia se ei korjaa.

2.3. Virtualisointi

2.3.1. Info

Virtualisoinnilla tarkoitetaan teknologiakokoelmaa, jolla pystytään ajamaan jotakin järjestelmää jonkin toisen järjestelmän sisällä. Esimerkiksi mikäli Digabi-käyttöjärjestelmä virtualisoitaisiin, olisi se käyttöjärjestelmä käyttöjärjestelmän sisällä. Samanlaista teknologiaa voidaan hyödyntää myös monissa muissa käyttötarkoituksissa, kuten silloin, kun jollakin tietokoneella ajetaan jollekin toiselle tietokoneelle tarkoitettua ohjelmaa.

2.3.2. Toteutus

Abitti-järjestelmää voitaisiin virtualisoida kätevästi esimerkiksi Oraclen avoimeen lähdekoodiin perustuvalla VirtualBox-nimisellä ohjelmalla kokelaiden koneilla. Tekninen toteutus voisi toimia esimerkiksi seuraavasti. Olisi olemassa jonkinlainen asennuspaketti, joka ladattaisiin esimerkiksi Digabin sivuilta. Asennuspaketti asentaisi ohjelman, joka huolehtii Digabi-käyttöjärjestelmän ja VirtualBox:in lataamisesta ja päivittämisestä. Lataaminen voisi tapahtua vaihtoehtoisesti joko torrent protocollan (vertaisverkko protocollan) tai http protocollan kautta riippuen siitä, kuinka paljon palvelinresursseja on käytettävissä. Yksi latausvaihtoehdoista tulisi olla myös itse koejärjestelmän palvelinkone suoraan, koska mikäli koko luokka alkaisi ladata päivitystä yhtä aikaa, rasittuisi koulun verkko tästä kohtuuttomasti ja esimerkiksi suljetussa ylioppilaskoeverkossa päivitysten lataaminen internetistä ei tulisi kysymykseen. Ohjelman tulisi myös osata säätää VirtualBoxin asetukset automaattisesti oikeiksi.

Virtualisointi aiheuttaa jo automaattisesti melko vahvan tietoturvan, mutta muutamista asioista olisi pakko huolehtia. Esimerkiksi VirtualBox pitäisi muokata sellaiseksi, että sen kokonäytöntilasta ei pysty poistumaan mitenkään VirtualBoxin ollessa päällä. Sen asetusvalikkoihin ei tulisi myöskään olla pääsyä millään kikalla. Lisäksi levykuva pitäisi varmistaa juuri ennen käynnistystä jollakin algoritmilla, kuten md5, jottei kukaan pääse peukaloimaan ympäristöä ennen käynnistystä. VirtualBoxia ei tulisi myöskään pystyä käynnistämään ilman käynnistyssovellusta, niin että esimerkiksi levykuvan tarkistaminen jäisi välistä.

Suuren osan tästä alla olevasta tekniikasta ei tulisi näkyä loppukäyttäjälle ollenkaan. Käyttäjän tehtäväksi tulisi ainoastaan osata asentaa ohjelmistokokonaisuus asennusohjelmasta

kerran ja sen jälkeen seuraavilla kerroilla kyetä painamaan yhtä painiketta, joka tarkistaisi päivitykset ja sen jälkeen käynnistäisi koko kokonaisuuden. Mahdollisesti käyttäjä tulisi velvoittaa myös laittamaan tietokoneen UEFI tai BIOS asetuksista laitteistotason virtualisointi päälle, mikäli mahdollista. Laitteistotason virtualisointia ei ole vaadittu VirtualBoxissa, mutta se nopeuttaa virtualisoidun käyttöjärjestelmän toimintaa.

Virtualisointiympäristöstä voisi myös tehdä erityisen version opettajille ja muille kokeenluojille. Kyseisessä versiossa olisi mahdollista ajaa Abitin serveriversiota ja kokelasversiota vaikkapa vierekkäisissä ikkunoissa. Näin kokeenluojat pääsisivät testaamaan saumattomasti kokeen toimivuutta ja varmistamaan helposti ennakkoon, että koe toimii, niin kuin pitääkin.

2.3.3. Hyvät puolet

Virtualisoinnin hyvänä puolena olisi helppokäyttöisyys loppukäyttäjille. Järjestelmän voisi käynnistää vaivattomasti ilman ainuttakaan muistitikkua suoraan tavalliselta työpöydältä nappia painamalla. Käynnistyksen helppous olisi täysin eri tasolla kuin tällä hetkellä ja järjestelmä mahdollistaisi pientenkin testien pitämisen sen nopeuden takia. Laitteiston epäyhteensopivuudet lakkaisivat olemasta heti, koska isäntäkäyttöjärjestelmä huolehtisi kaikista ajureista, kuten tietokonetta tavallisestikin käytettäessä.

Tietoturva virtualisoinnissa olisi oikein toteutettuna lähes samalla tasolla kuin muistitikuilta käynnistettäessä, jollei jopa parempi. Virtualisoinnissa vilppiä voisi nimittäin tutkia kahdelta taholta, niin isäntäkäyttöjärjestelmän, kuin Digabi-käyttöjärjestelmänkin taholta. Järjestelmä voitaisiin tarkistaa ennen käynnistystä helposti toisin kuin muistitikulta käynnistäessä. Lisäksi jo käynnissä olevan Virtualisointiohjelman hakkerointi on lähes mahdotonta, jollei siinä ole tunnettuja bugeja ja niitä tuskin on, jos VirtualBox vain pidetään päivitettynä. Hyvän suojauksen nimittäin jo käynnissä olevalle virtualisoidulle käyttöjärjestelmälle tarjoaa nykyisten käyttöjärjestelmien käyttämät monimutkaiset muistinsuojausmekanismit.

2.3.4. Huonot puolet

Virtualisoinnin selkein huono puoli on se, että se vie tehoa ja tilaa kokelaiden koneilta. Kokonaisuus tuskin tulisi viemään tilaa aivan merkittävästi. Koko systeemin pitäisi mahtua helposti

alle viiteen gigatavuun. Virtualisoinnin vaatima isompi tietokoneen teho taas on mutkikkaampi ongelma. Monet kannettavat pystyvät nykyisellään pyörittämään virtuaalista käyttöjärjestelmää, mutta eivät kuitenkaan tällä hetkellä kaikki ainakaan riittävän nopeasti. Ratkaisu tähän on aika. Digabin sisältämät avoimen lähdekoodin ajurit toimivat nykyisellään useimmiten hyvin vanhempien kannettavien koneiden kanssa, koska kehittäjillä on ollut kauemmin aikaa kehittää ajureita vanhempiin tietokoneisiin, kun taas uudempiin koneisiin ei ole vielä ehditty tehdä yhteensopivia ajureita. Virtualisointi ei estä sitä, etteikö tikulta käynnistämistä voitaisi edelleen hyödyntää myös vaihtoehtoisesti. Kun viimeistenkin käyttäjien tietokoneet ovat uudistuneet jossain kohtaa tarpeeksi voidaan tikulta käynnistämisestä luopua kokonaan.

Työmäärä on toinen ongelma. Virtualisointijärjestelmä vähentää toisaalta töitä, kun ei enää tarvitse huolehtia siitä, että kaikki koneet varmasti toimisivat, mutta toisaalta uudesta ohjelmasta tulisi luoda kolme pakettia niin Windowsille, Macille kuin Linuxille. Tämä lisää selvästi töitä, kun pitää rakentaa kolme eri versiota. Lisäksi Linuxin tilanne on erittäin hankala distrojen monimuotoisuuden takia. Linuxilla tilanne onkin käytännössä se, että omat versiot kannattaisi rakentaa ainoastaan kaikista yleisimmille distroille, kuten vaikkapa Ubuntulle ja Debianille.

2.3.5. Käyttäjäkunta

Virtualisointijärjestelmän luominen hyödyttäisi kaikkia käyttäjiä. Tikkuja ei tarvitsisi enää tehdä ainakaan enää niin paljon kuin ennen. Koejärjestelmä olisi käytännöllisempi ja monikäyttöisempi. Lisääntyneen käytännöllisyyden takia kokeita voitaisiin järjestää aiempaa enemmän ja tulevilla ylioppilaskokelailla olisi hyvin aikaa tutustua järjestelmän käyttöön ja toimintaan lukiovuosien aikana. Kokeen laatijan työstä tulisi yksinkertaisempi.

2.3.6. Kustannukset

Virtualisointi aiheuttaisi koejärjestelmän kehittäjille jonkin verran lisäkustannuksia ohjelmiston kehittämiseen menevän rahan takia, mutta toisaalta se vähentäisi paljon rahanmenoa järjestelmän toisiin osiin. Myös ohjelmiston latauspalvelimen ylläpito olisi jatkuva rasite, mutta kuluja voisi pienentää huomattavasti torrentin avustuksella, jolloin kaikki päivittäjät lähettäisivät päivitystä samalla toisilleen.

2.3.7. Bonus

Virtualisointijärjestelmän kriittisten osien, kuten erilaisten käyttöjärjestelmäpakettien, valmistuttua, olisi mahdollista tietoturvaa ja toiminnallisuutta vielä tehostaa. Esimerkiksi VirtualBox:in käynnissä olevia käyttöjärjestelmiä pystyy ohjaamaan etänä RDP (Remote Desktop Protocol) avulla. Etäohjausta ei sinällään koetilanteissa tarvita, mutta järjestelmällä pystyisi vakoilemaan kokeentekijöiden näyttöä kokeen aikana pistos tyyppisesti, mikä parantaisi turvallisuutta.

Työteliään järjestelmästä tekee se, että silloin kaikkien etäkatseltavien koneiden IP-osoitteet (Internet Protocol) on oltava jotenkin helpossa listassa, josta voi vain klikata yhteyksiä auki. Lisäksi koneet pitää olla jotenkin tunnistettavissa toisistaan. Ratkaisu ensimmäiseen ongelmaan voisi olla se, että palvelinkone ylläpitää itse DHCP-ohjelmistoa ja verkkoinfrastruktuuri koostuu pelkästään kytkimistä. Näin IP-osoitteet olisivat suoraan palvelinkoneen tiedossa ja hallinnassa. Toisaalta palvelinkone voisi myös etsiä koneiden IP-osoitteet suoraan verkosta, mutta tämä tarvitsisi myös oman järjestelmänsä.

Koneiden tunnistaminen voisi tapahtua niin, että oma nimi tulisi syöttää ohjelmaan, joka käynnistää virtuaalikoneen. Tämän jälkeen ohjelma lähettäisi tiedon palvelinkoneelle. Toinen yksinkertaisempi vaihtoehto olisi se, että nimi tulisi syöttää virtuaalikoneeseen kirjautumisen jälkeen ja se näkyisi aina jossakin työpöydän nurkassa. Tällöin kovin erityisiä järjestelyjä ei vaadittaisi, vaan tietokoneita etänä valvova henkilö näkisi nimen aina yhteyden avatessaan.

2.3.8. Lopputulema

Virtualisoinnin avulla saavutettaisiin kiosk-konetta lähimpänä oleva turvallinen järjestelmä, joka olisi joka tasolta täysin hallinnassa. Tekniikassa on omat haasteensa, mutta lopputuloksen pitäisi olla hyvä ja pitkälle kestävä.

3. Virtualisoinnin tulevaisuuden näkymiä

3.1. Infoa

Kaikki tässä osassa oleva on jo nykyään mahdollista, mutta vaatisi nykyisessä Infrastruktuurissa laajoja muutoksia niin verkon, kun laitteidenkin puolesta. Nykyisen hinnan takia tämä voidaan laskea vain pitemmän ajan suunnitelmaksi.

3.2. Mahdollisuudet

Jos koejärjestelmän kehitystä jatketaan virtualisoinnin suuntaan, on mahdollista, että koko koejärjestelmää pidetään yllä ainoastaan yhdellä tehokkaalla palvelinkoneella, johon kaikki kokelaiden laitteet ovat yhteydessä joko langallisesti tai langattomasti. Tässä mallissa kokelaiden omat laitteet pyörittäisivät ainoastaan yhtä etäkäyttöohjelmaa ja toimisivat ikään kuin näyttöinä sillä aikaa, kun kaikki laskenta tapahtuisi suoraan palvelimella.

Tässä mallissa hienous olisi se, että kaikki laitteet, jopa kännykät ja tabletit mukaan lukien, voisivat toimia näyttöinä. Tällä tavalla kaikista laitteista saataisiin täysin yhdenvertaisia ja lukiolaiset voisivat huoletta ostaa minkälaisen laitteen tahansa. Lisäksi kun kaikki kokelaat työskentelisivät yhden palvelimen alla voisi valvoja tarkkailla kaikkien kokelaiden näyttöjä reaaliajassa palvelinkoneen näytöltä vastaavalla tavalla, kuten 2.3.6. Bonus kohdassa on esitetty.

3.3. Toteutus

Ympäripyöreästi kuvattuna järjestelmä vaatisi mahdollisimman tehokkaan palvelinkoneen tai jopa useampia. Palvelinkoneen tulisi pitää yllä monia virtuaalikoneita, joihin pääsisi sisään etäyhteyden kautta. Esimerkiksi VirtualBox-ohjelman pitäisi sisältää kaiken mahdollisen tähän tarkoitukseen vaadittavan. Sillä voi ajaa monia virtuaalikoneita yhtä aikaa ja sillä pystyy jakamaan liveyhteyksiä RDP:n kautta. Tällaiseen Virtuaali-istuntoon kokelas pääsisi asentamalla RDP-yhteysohjelman ja saamalla kokeen järjestäjältä yksityisen satunnaisesti arvotun portinnumeron.

3.4. Ongelmia

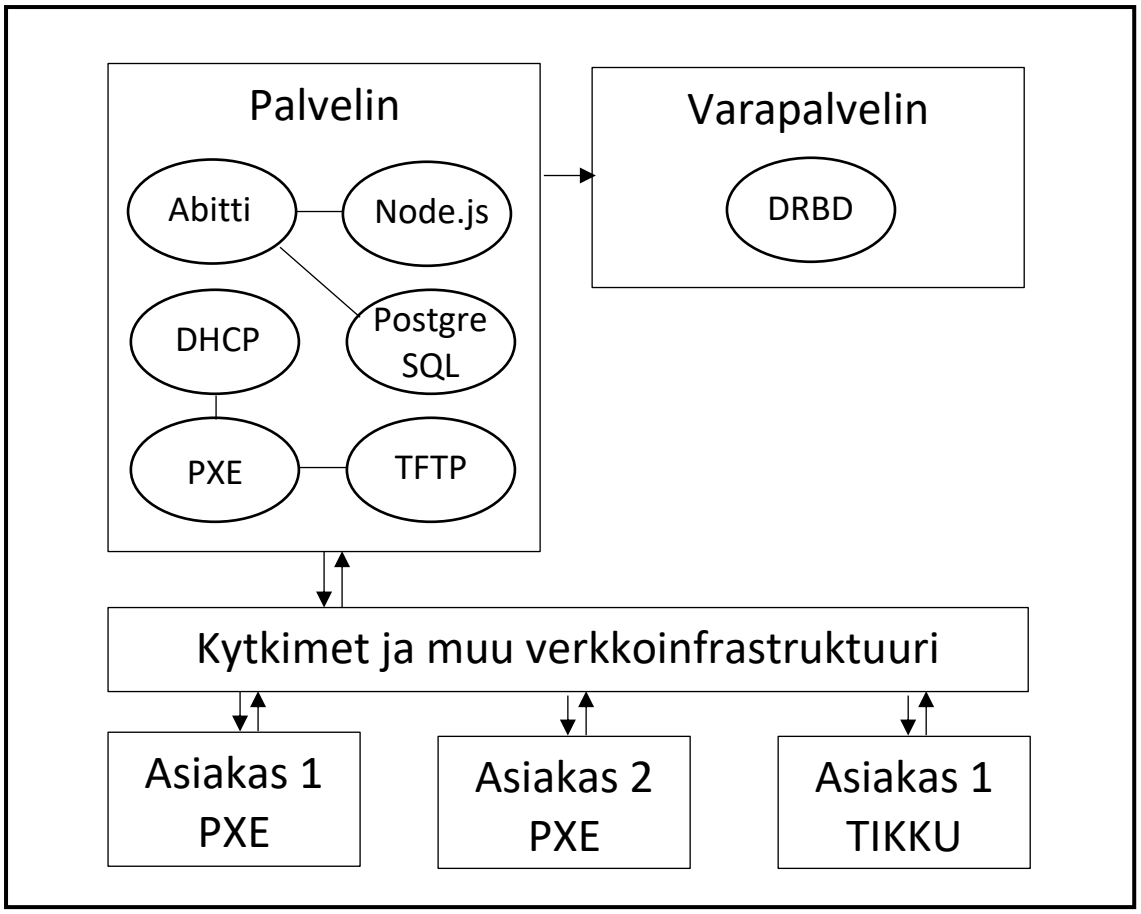
Palvelimessa tarvitsisi olla yksinkertaisesti huomattavan paljon tehoa. Levykäytön takia palvelinkäyttöjärjestelmä olisi käytännössä pakko asentaa suoraan tietokoneen kiintolevylle. Tämäntyyppiseen asennukseen taas olisi kehiteltävä tehokas päivitysjärjestelmä. Päivityksen tarpeen voisi kiertää siten, että virtuaalityöpöytiä tarjoava palvelin toimisi erillisellä tietokoneella esimerkiksi vaikkapa Windowsin päällä ja virtuaalikäyttöjärjestelmän levykuva päivitettäisiin samantapaisella käynnistysohjelmalla kuin kokelaiden tietokoneet.

3.5. Miten edetä?

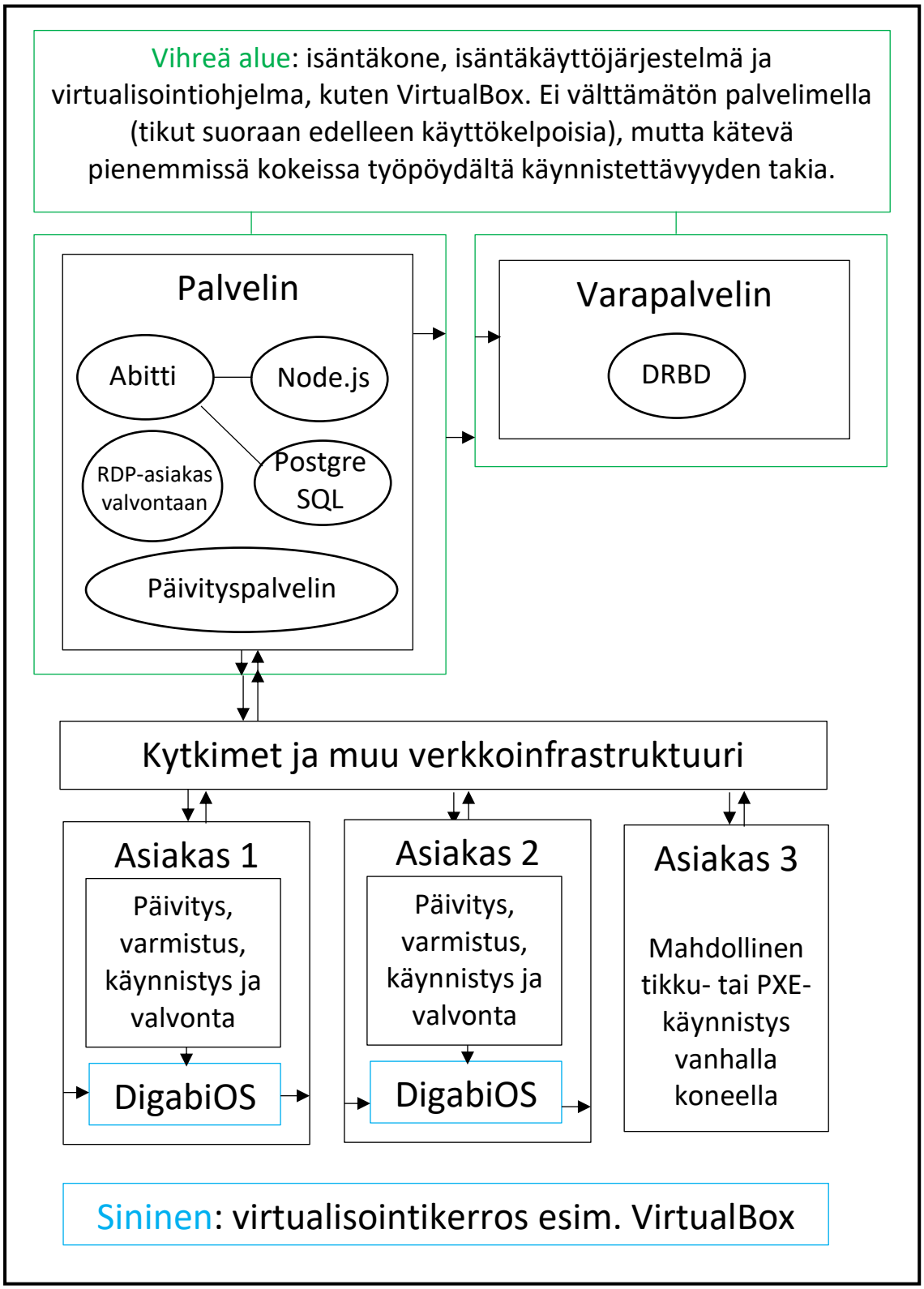
Etävirtuaaliyhteysjärjestelmää voisi testata aluksi vain pienessä mittakaavassa vaikkapa laitteille, joita ei tällä hetkellä tueta, kuten ARM-prosessorilla varustetut tabletit. Näin heti ei tarvittaisi todella tehokkaita palvelinkoneita, kun käyttäjämäärät pysyisivät vielä pieninä. Tämänkaltaisen pienimittaisen kokeilun voisi aloittaa jo lähivuosina.

4. Kaavioita

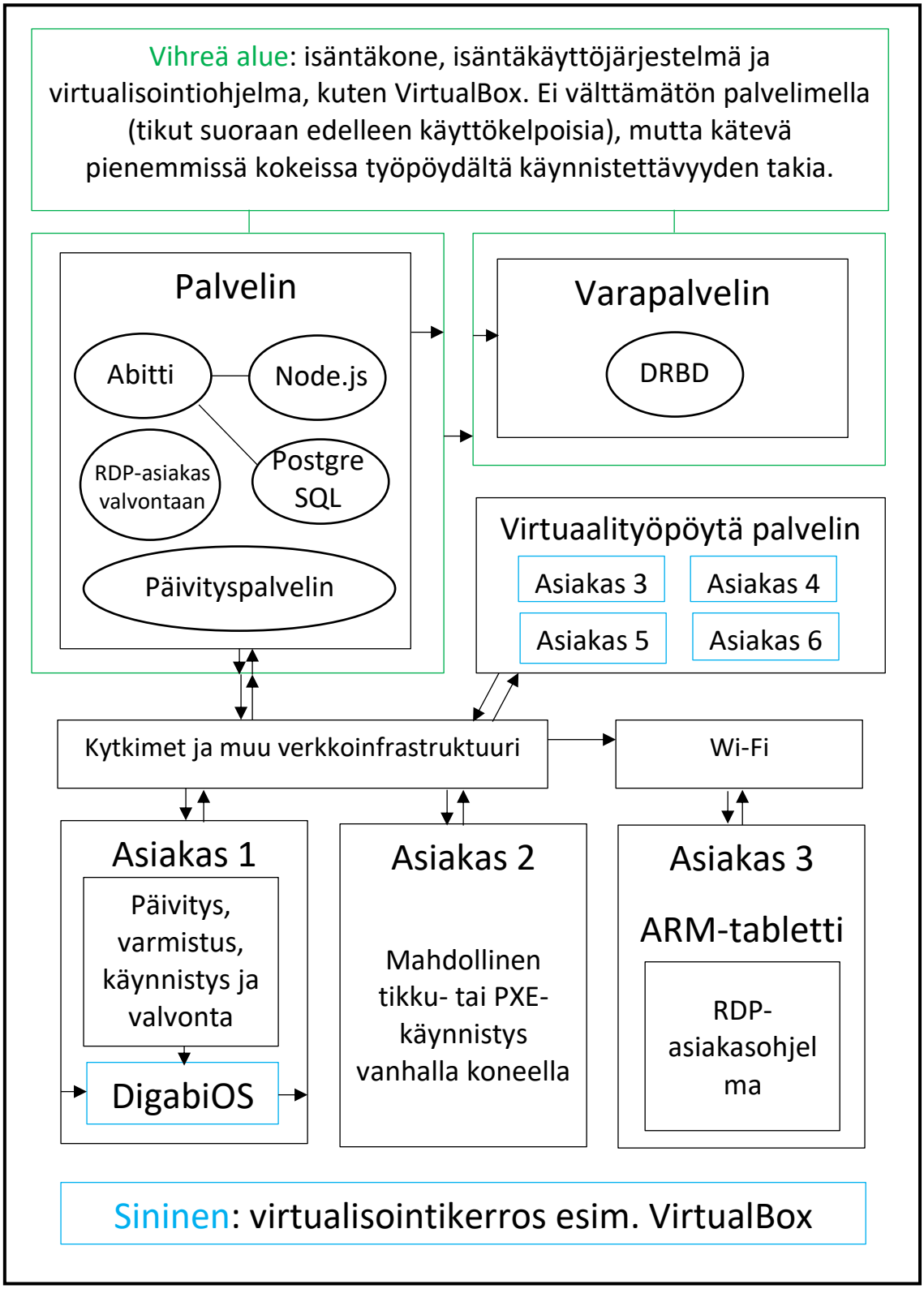
4.1. PXE



4.2. Yksinkertainen virtualisointi



4.3. Monimutkainen virtualisointi



Virtuaalityöpöytäpalvelin voidaan jakaa myös useammalle tietokoneelle, mutta toisaalta integroida myös muista koejärjestelmistä huolehtivaan palvelimeen.

5. Loppusanat

PXE-käynnistyksen toteuttamisella kannattaisi mahdollisesti aloittaa, koska se eliminoisi tikkujen käyttöön liittyviä tietoturvaongelmia. Pitemmällä tähtäimellä kuitenkin pitäisi siirtyä kohti virtualisointisuunnitelmia, koska se on käytännössä ainut mahdollisuus tehdä täysin yhtäläinen, turvallinen ja helppo kokemus kaikilla laitteilla. Virtualisointi mahdollistaa erittäin kontrolloitavan hiekkalaatikon, joka ei kuitenkaan tule haittaamaan tietokoneen käyttöä muussa käytössä juuri millään tavalla. Lisäksi nykyisestä työstä ei mene palaakaan hukkaan, vaan kaikki tehty on edelleen hyödynnettävissä. Nyt on enää luotava uusi ympäristö, jossa jo kaikkea tehtyä voidaan hyödyntää.

Virtualisointi on lisäksi käytännössä ikivihreä ratkaisu, koska vaikka laitteisto ja käyttöjärjestelmät vaihtuvat, niin ajan myötä tulevat erilaiset virtualisointiohjelmat toimimaan aina jollakin tavalla. Ilman tietokoneohjelmien ja käyttöjärjestelmien virtualisointia ei monia vanhempia ohjelmia ja järjestelmiä tälläkään hetkellä pystyisi käyttämään.

Virtualisoidulla tai osaksi virtualisoidulla ympäristöllä pystytään toteuttamaan äärimmäisen joustava järjestelmä, jossa monilla eri tavoilla toimivat laitteet voivat silti tarjota yhtenevän käyttäjäkokemuksen. Kustannukset pysyisivät niin kauan minimissään, kun mahdollisimman monet pystyisivät tuomaan laitteita, jotka pystyvät itsenäisesti ylläpitämään järjestelmää. Virtualisoidun ympäristön suurena etuna olisi kuitenkin se, että se mahdollistaisi aivan kaikenlaisen laitteiden hyödyntämisen tarvittaessa ja resurssien riittäessä.