

**MIRADORE MANAGEMENT SUITE -JÄRJESTELMÄN KÄYTTÖ LINUX  
DEBIAN -TYÖASEMIEN JA -PALVELIMIEN ASENNUKSIIN SULJETUSSA  
YMPÄRISTÖSSÄ**



Ammattikorkeakoulututkinnon opinnäytetyö

Tieto- ja viestintäteknikka, Riihimäki

Syksy 2021

Ari Mustonen

## TIIVISTELMÄ

Opinnäytetyön tilaaja käyttää Miradore Management Suite -järjestelmää päätelaitteiden (työasemat, palvelimet) käyttöjärjestelmien asentamiseen. Työn tavoitteena oli muokata järjestelmää siten, että sitä käyttämällä voidaan asentaa Linux Debian -pätelaitteita tilaajan vaatimukset täyttävällä tavalla.

Työ aloitettiin asentamalla tilaajan järjestelmää vastaava demo-ympäristö. Järjestelmän toimintaa tutkittiin demo-ympäristössä ja tutkimuksen ohessa kerättiin tietoa järjestelmän toiminnasta. Kerättyyn tietoon perustuen tehtiin suunnitelmat järjestelmän muokkaamisesta työn tilaajan vaatimusten mukaiseksi.

Järjestelmään tehdyt muutokset toteutettiin vaiheittain ja jokaisen vaiheen päätteeksi testattiin järjestelmän toimintaa asentamalla päätelaitteita järjestelmää käyttäen. Tähän työhön on dokumentoitu järjestelmään tehdyt muutokset.

Kaikki työn tilaajan vaatimukset saatiin toteutetuiksi järjestelmään ja muokattu järjestelmä on jo tuotantokäytössä. Ongelmia ei juurikaan esiintynyt, jos ei sellaiseksi lasketa työn tekijän vähäistä kokemusta VBScript-ohjelmointikielestä työtä aloitettaessa.

Avainsanat Linux, asennus, hallintajärjestelmät, automaatio

Sivut 32 sivua ja liitteitä 27 sivua

---

Author Ari Mustonen

Year 2021

Subject Using Miradore Management Suite to install Linux Debian workstations and servers in a closed environment

Supervisors Teemu Järvenpää

---

ABSTRACT

The company that commissioned the thesis is using Miradore Management Suite system for installing the operating systems of the devices (workstations, servers). The aim of the work was to modify the system so that it could be used to install Linux Debian devices in a way that meets the requirements of the company.

The work was started by installing a demo environment corresponding to the system the company uses. The demo environment was used to study the operations of the system and for collecting information from the system. Based on the collected information, plans were made to modify the system to meet the requirements.

The changes made to the system were implemented in stages. At the end of each stage, the operation of the system was tested by installing devices. Modifications to the system have been documented in this thesis.

All the company's requirements were filled. The modified system is already in production use. There were few problems, unless the author's limited experience with the VBScript programming language in the beginning of the project could be counted as one.

Keywords Linux, installation, management systems, automation

Pages 32 pages and appendices 27 pages

## Sisällys

1	Johdanto .....	1
1.1	Usean erilaisen vastaustiedoston käyttäminen asennuksissa .....	1
1.2	Omien pakettivarastojen lisääminen järjestelmään .....	2
1.3	Käyttäjähallinnan piilottaminen loppukäyttäjiltä .....	2
2	Kehittämistyön tietoperusta .....	3
2.1	Miradore Management Suite.....	3
2.2	Järjestelmän komponentit .....	4
2.2.1	Miradore-palvelin.....	4
2.2.2	Miradore-asennuspiste .....	4
2.2.3	Miradore-asiakassovellus.....	6
2.3	Järjestelmän käyttämät verkkopalvelut ja -protokollat.....	6
2.3.1	DHCP .....	6
2.3.2	PXE.....	7
2.3.3	TFTP .....	7
2.3.4	WinPE .....	8
2.3.5	Windows-levyjako .....	9
2.3.6	HTTP-tiedostojako .....	9
2.3.7	Linux Debian -pakettivarasto (repository) .....	10
2.4	Asennusprosessi laitetta asennettaessa .....	10
3	Kehittämistyön tavoite ja tarkoitus.....	12
4	Suunnittelu .....	12
4.1	Järjestelmästä muokataan ainoastaan Debian.vbs –tiedostoa .....	15
4.2	Asennuksen vastaustiedostosta luodaan pohjatiedosto .....	15
4.3	Jälkiasennus-skripteistä luodaan pohjatiedostot .....	15
5	Toteutus.....	17
5.1	Pohjatiedostojen luominen.....	17
5.2	Oman pakettivaraston (repository) lisääminen.....	20
5.3	Jälkiasennus-skriptien muokkaaminen .....	23
5.4	Lokalisointi .....	25
5.5	Automaattisen asennuksen vastaustiedoston käsittely .....	28
6	Johtopäätökset ja pohdinta.....	31
	Lähteet.....	32

## Kuvat, taulukot ja kaavat

Kuva 1. Miradore-järjestelmän rakenne.....	3
Kuva 2. Asennuspisteiden hierarkia. (Miradore, 2020, s. 919) .....	5
Kuva 3. Linux Debian -laitteen asennusprosessi. ....	11
Kuva 4. Demo-ympäristö .....	12
Kuva 5. Miradore-palvelimen asennusvalinnat.....	13
Kuva 6. Asennushakemiston sisältö asennuspisteellä .....	17
Kuva 7. Laiterooli (DEBIAN) Miradore-järjestelmässä.....	18
Kuva 8. Pakettivaraston julkaiseminen ja allekirjoittaminen .....	22
Kuva 9. Järjestelmään määritetty sijainti (Location) .....	26
Kuva 10. Salasanatiedoston luominen ja nimeäminen .....	27
Taulukko 1. Miradore-järjestelmän käyttämät DHCP-optiot. (Miradore, 2020, s. 787) ...	7

## Liitteet

Liite 1	Alkuperäinen Debian.vbs -tiedosto
Liite 2	Alkuperäinen preseed.cfg -tiedosto
Liite 3	Alkuperäinen latecmd.sh -tiedosto
Liite 4	Alkuperäinen postinst.sh -tiedosto
Liite 5	Pakettivaraston allekirjoitusavaimien luominen
Liite 6	Shell-skripti pakettivaraston julkaisemiseen
Liite 7	Shell-skripti pakettivaraston julkaisutiedostojen allekirjoittamiseen
Liite 8	Shell-skripti salasanatiedoston luomiseen
Liite 9	Lopullinen Debian.vbs -tiedosto
Liite 10	Lopullinen preseed.cfg -pohjatiedosto
Liite 11	Lopullinen latecmd.sh -pohjatiedosto
Liite 12	Lopullinen postinst.sh -pohjatiedosto

## 1 Johdanto

Työn tilaaja (myöhemmin tilaaja) käyttää Miradore Management Suite (myöhemmin Miradore) -järjestelmää työasemien ja palvelimien käyttöjärjestelmien asennuksiin. Työn tekijä työskentelee tilaajan palveluksessa ja työtehtäviinsä liittyen ylläpitää ja hallinnoi kyseistä järjestelmää, sekä osallistuu vahvasti myös järjestelmän käyttöön liittyvään tutkimus- ja kehittämistyöhön.

Tilaaja on valmistelemassa Linux Debian 10 -käyttöjärjestelmän käyttöönottoa ja kyseistä käyttöjärjestelmää käyttävien laitteiden asennukset on tarkoitus automatisoida Miradore-järjestelmää käyttäen.

Miradore-järjestelmä tukee Linux Debian 10 -käyttöjärjestelmän asentamista. Tilaajalla on kuitenkin asennuksiin liittyen järjestelmän osalta sellaisia vaatimuksia, joita järjestelmä ei tue. Tämän työn tarkoituksena on tutkia mahdollisuuksia muokata Miradore-järjestelmää vastaamaan tilaajan tarpeita.

### 1.1 Usean erilaisen vastaustiedoston käyttäminen asennuksissa

Miradore-järjestelmä luo Linux Debian -käyttöjärjestelmän asennuksen yhteydessä automaattisen asennuksen vastaustiedoston (preseed). Vastaustiedosto on sama kaikille Debian-asennuksille, eikä käyttöjärjestelmäkokoontaan voi vaikuttaa järjestelmästä tehtävillä valinnoilla.

Työn tavoitteena on tuottaa järjestelmän oletusasetuksista poikkeava malli, jossa samasta käyttöjärjestelmä-repositorystä pystytään asentamaan erilaisia käyttöjärjestelmäkokoontajia (työasema, palvelin) automaatisoimalla asennus järjestelmästä tehtävillä valinnoilla ja muokkaamalla asennukseen käytettävää vastaustiedostoa (preseed) valintoja vastaavaksi.

## **1.2 Omien pakettivarastojen lisääminen järjestelmään**

Miradore-järjestelmä käyttää Linux Debian -asennuksissa vain yhtä pakettivarastoa (repository).

Työn tavoitteena on lisätä järjestelmään erillinen pakettivarasto työn tilaajan omille sovelluksille, allekirjoittaa se omalla avaimella ja saattaa se asennuksen kohteena olevien käyttöjärjestelmien käyttöön jo käyttöjärjestelmän asennusvaiheessa.

## **1.3 Käyttäjähallinnan piilottaminen loppukäyttäjiltä**

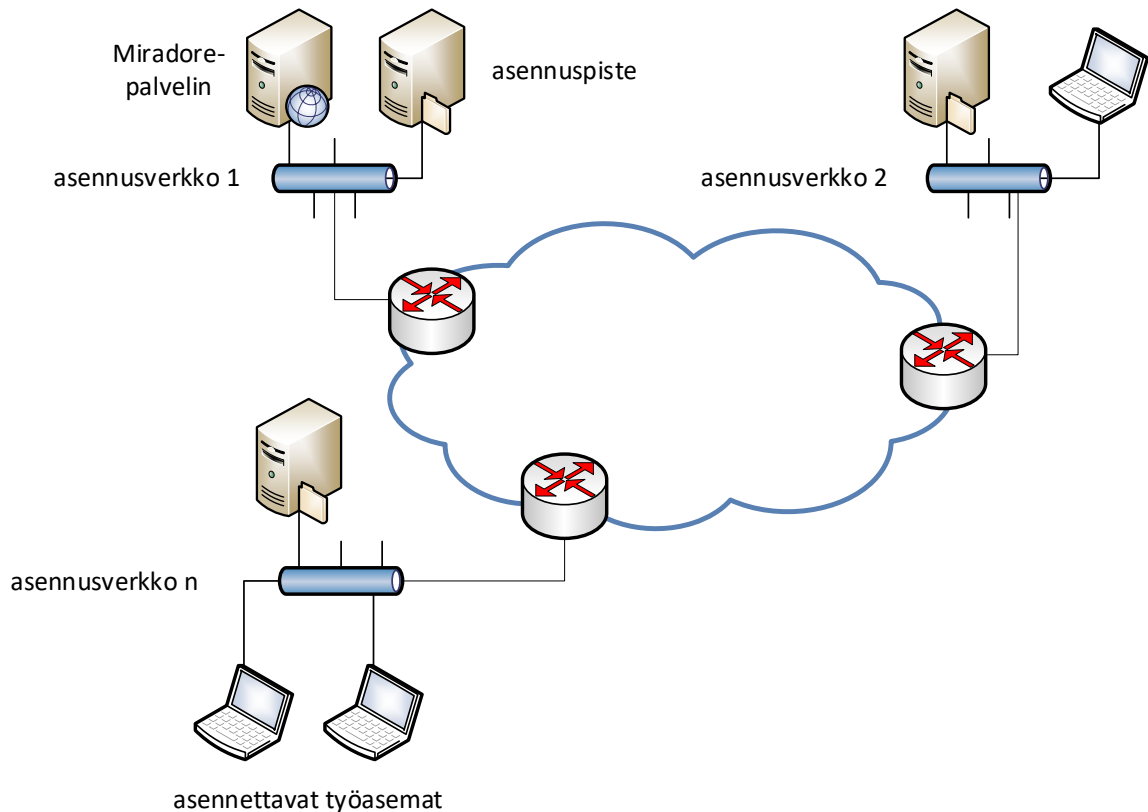
Miradore-järjestelmällä tehtävä Linux Debian -työaseman (ja palvelimen) käyttäjähallinta tehdään käyttöjärjestelmän jälkeen asennettavilla sovelluspaketeilla. Nämä sovelluspaketit ovat järjestelmän loppukäyttäjille näkyvä komponentti.

Työn tavoitteena on tutkia mahdollisuutta sisällyttää käyttäjätunnusten ja salasanojen hallinta osaksi käyttöjärjestelmän asennusta ja piilottaa näin käyttäjätunnuksiin liittyvä hallinnointi järjestelmän loppukäyttäjiltä.

## 2 Kehittämistyön tietoperusta

Tilaaaja on käyttänyt Miradore-järjestelmää Windows-asennuksiin jo yli kymmenen vuoden ajan ja tutkimustyön tekijä on toiminut järjestelmän loppukäyttäjänä ja ylläpitäjänä koko tuon ajan. Periaatekuva tilaajan Miradore-järjestelmästä on esitettyä kuvassa 1.

Kuva 1. Miradore-järjestelmän rakenne.



### 2.1 Miradore Management Suite

Miradore on hallintajärjestelmä IT-ympäristöille, jotka sisältävät tietokoneita, palvelimia, tabletteja, älypuhelimia ja POS-laitteita. Miradoressa eri laitteisto- ja ohjelmistoalustojen hallintaprosessit on integroitu saumattomasti yhtenäiseksi hallintaratkaisuksi, joka tukee monenlaisia laitealustoja, mukaan lukien Linux. Kun Miradore sisällytetään asiakkaan prosesseihin, sen avulla voidaan luoda automaattisia toimintoja välittömän arvon luomiseksi. (Miradore, n.d.)



Yksi monista järjestelmän toiminnoista on käyttöjärjestelmien ja sovellusten asennusten automatisointi ennalta määritettyjen asetusten mukaan. Tässä työssä tutkitaan Linux Debian -käyttöjärjestelmän ja sovellusten asentamista työn tilaajan vaatimusten mukaisesti.

## **2.2 Järjestelmän komponentit**

Miradore-järjestelmä koostuu useista komponenteista. Seuraavassa lueteltuna tämän tutkimuksen kannalta keskeiset järjestelmän osat.

### **2.2.1 Miradore-palvelin**

Miradore-palvelin sisältää kaikki järjestelmän hallintatiedot. Palvelin toimii Windows Server -käyttöjärjestelmässä. Palvelimen ydin on keskitetty tietokanta, johon kaikki tiedot tallennetaan. Tietokantana käytetään Microsoft SQL Server (express) -tietokantaa. Järjestelmää käytetään verkkopohjaisen graafisen käyttöliittymän kautta. Järjestelmän web-käyttöliittymä ja hallintakonsoli toimivat Microsoft IIS:ssä (Internet Information Services). Pienissä ympäristöissä järjestelmäkokonaisuuteen tarvitaan vain yksi palvelin, mutta suurempiin järjestelmiin voidaan asentaa erillinen tietokantapalvelin. (Miradore, 2020, ss. 90, 92)

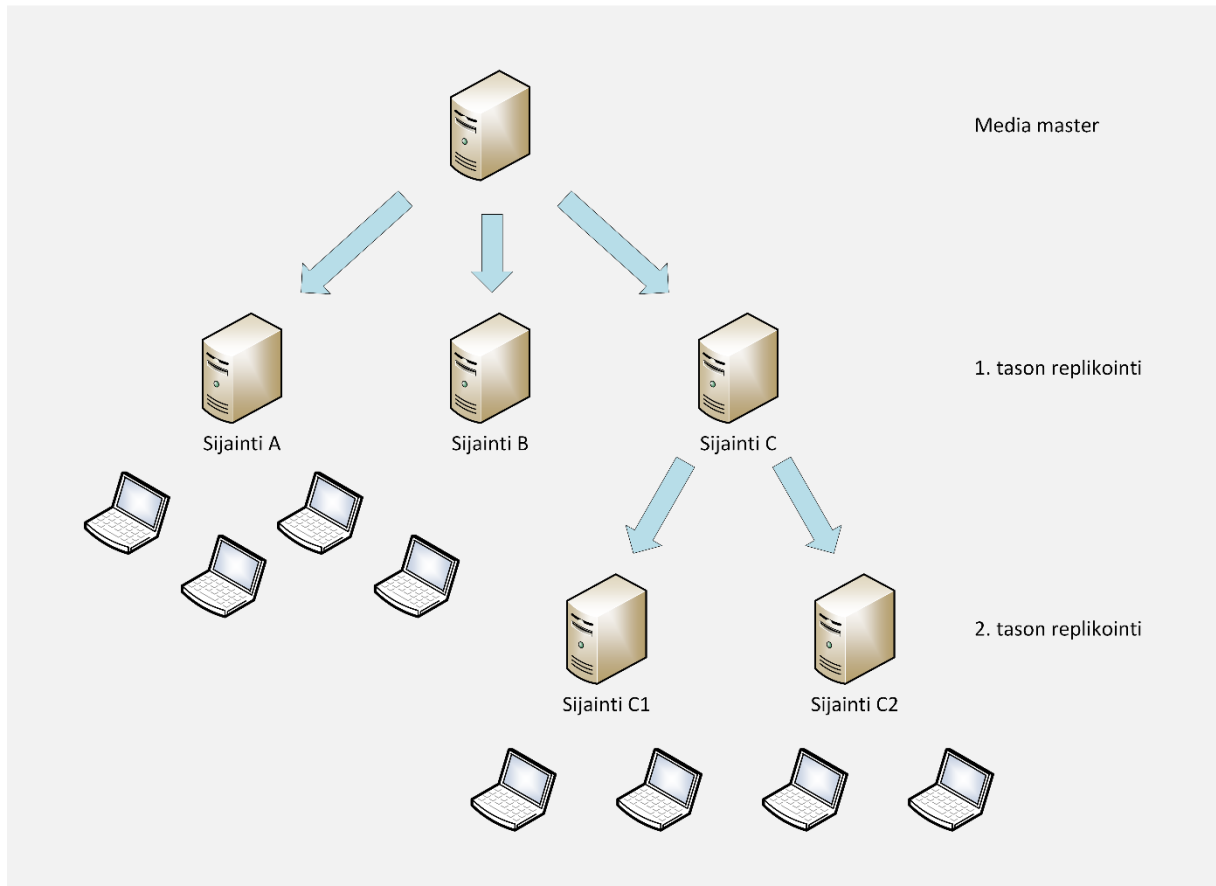
Tilaajan järjestelmässä Miradore-palvelin on asennettu virtualisoituun Windows Server 2016 -käyttöjärjestelmään. Microsoft SQL Server -tietokantapalvelin toimii samassa virtuaalipalvelimessa.

### **2.2.2 Miradore-asennuspiste**

Asennuspisteellä sijaitsee käyttöjärjestelmien ja sovellusten asennuksiin käytettävä data. Järjestelmä voi sisältää useita asennuspisteitä, joista valitaan aina automaattisesti lähin. Jos järjestelmää käytetään vain yhdessä lähiverkossa (LAN) tulisi harkita vain yhden asennuspisteen käyttöä. Mikäli asennuksia tehdään useissa eri sijainneissa (verkoissa) on tarpeellista rakentaa järjestelmään sijainteja vastaava asennuspisteiden hierarkia (Kuva 2). Hierarkian avulla optimoidaan verkkoliikennettä mahdollisimman lähelle asennettavia laitteita. Yksi järjestelmän asennuspisteistä on ns. pääasennuspiste, johon tehtävät

muutokset replikoidaan automaattisesti muille asennuspisteille. (Miradore, 2020, ss. 90, 101)

Kuva 2. Asennuspisteiden hierarkia (Miradore, 2020, s. 919).



Asennuspiste on siis palvelin, josta jaetaan asennusten tarvitsemaa dataa. Data jaetaan HTTP-jakona (Linux-asennukset) ja Windows-verkkojen tiedostojakona (Windows-asennukset). Lähtökohtaisesti asennuspiste sijaitsee aina asennettavan päätelaitteen kanssa samassa lähiverkossa, koska asennusten yhteydessä siirrettävät datamäärät voivat olla hyvinkin suuria.

Tilaajan järjestelmässä asennuspisteet ovat virtualisoituja Linux Debian -palvelimia. Asennuspisteisiin on konfiguroituna tiedostojakojen lisäksi myös muut asennuksen tarvitsemat palvelut.

### **2.2.3 Miradore-asiakassovellus**

Asiakassovellus (client) asennetaan hallittaviin laitteisiin käyttöjärjestelmän asentamisen jälkeen. Sovellus kommunikoi palvelimen kanssa ja on vastuussa laitteessa suoritettavista ylläpitotehtävistä, kuten sovellusten asentamisesta. (Miradore, 2020, s. 107)

## **2.3 Järjestelmän käyttämät verkkopalvelut ja -protokollat**

Miradore-järjestelmä tarvitsee toimiakseen joitakin verkkopalveluita. Tilaajan järjestelmässä nämä palvelut tuotetaan asennuspisteenä toimivalla Linux Debian -palvelimella.

### **2.3.1 DHCP**

DHCP (Dynamic Host Configuration Protocol) on protokolla, joka mahdollistaa IP-osoitteiden ja muiden verkkoasetusten automaattisen konfiguroinnin. DHCP perustuu yleisimmin RFC-dokumentteihin 2131 ja 2132. DHCP toimii asiakas-palvelin -periaatteella. Asiakaslaitteet hakevat käynnistyessään konfiguraatitiedot verkossa olevalta DHCP-palvelinlaitteelta. DHCP-standardikehyksen lisäksi tietojen antamiseen voidaan käyttää myös laitevalmistajille varattua optiokenttää. (Kaario, 2002, s. 66)

Miradore-järjestelmä käyttää DHCP-protokollan optiokenttää välittämään tarvittavia tietoja asennettaville päätelaitteille. Järjestelmän käyttämät optiot ovat listattuina taulukossa 1.

Taulukko 1. Miradore-järjestelmän käyttämät DHCP-optiot (Miradore, 2020, s. 787).

Nro	Nimi	Selite
230	Miradore server	Miradore-palvelimen nimi tai IP-osoite
231	Miradore instance	Miradore-instanssin nimi
232	HTTP/HTTPS port	Miradore-käyttöliittymän HTTP/HTTPS –portti (1-65536) (oletus=80)
233	Use HTTPS	HTTPS päälle/pois yhdistettäessä Miradore-palvelimeen (0/1) (oletus=0)
234	WinPE keyboard layout	WinPE:n näppäimistöasettelu MS Windows lokalisointimuodossa (0409:0000xxxx)
235	Proxy server[:port]	Välityspalvelimen nimi tai IP-osoite (optiona porttinumero kaksoispisteellä erotettuna)
236	Proxy authentication method	Välityspalvelimen autentikointimenetelmä (none, basic, digest, ntlm)
237	Proxy user	Käyttäjänimi välityspalvelimeen kirjautumiseksi
238	Proxy user password	Salasana välityspalvelimeen kirjautumiseksi

### 2.3.2 PXE

PXE (Preboot eXecution Environment) on protokolla, joka mahdollistaa asiakaslaitteen käynnistämisen verkosta ladattavalla käyttöjärjestelmällä. PXE on määritetty laajasti käytössä olevien DHCP- ja TFTP-protokollien pohjalta. Asiakaslaite aloittaa protokollan DHCPDISCOVER-pyynnön yhteydessä ja jos DHCP-palvelin tukee PXE-protokollaa, se lähettää asiakaslaitteelle tiedon käynnistyspalvelimesta ja -tiedostosta. Asiakaslaite lataa TFTP-protokollaa käyttäen käynnistystiedoston TFTP-palvelimelta ja suorittaa tiedoston. (Intel Corporation, 1999, s. 10)

Miradore-järjestelmä käyttää PXE-protokollaa asiakaslaitteiden käyttöjärjestelmien asennuksien aloittamiseen. Protokollan avulla asiakaslaite käynnistetään WinPE-käyttöjärjestelmään ja käyttöjärjestelmässä suoritetaan Miradoren oma sovellus, jonka avulla tehdään asennuksen vaatimat määritykset ennen varsinaisen käyttöjärjestelmän asentamista.

### 2.3.3 TFTP

TFTP (Trivial File Transfer Protocol) on yksinkertainen protokolla, jota käytetään tiedostojen siirtoon. Protokolla on suunniteltu pieneksi ja helposti käyttöön otettavaksi. Tästä johtuen siitä puuttuu suurin osa normaalin FTP-protokollan ominaisuuksista. Ainoa asia, johon TFTP-

protokolla pystyy, on tiedostojen lukeminen ja kirjoittaminen palvelimelta tai palvelimelle. TFTP-protokolla ei tue hakemistoja tai käyttäjien autentikointia. (Network Working Group, 1992, s. 2)

Miradore-järjestelmä käyttää TFTP-protokollaa käynnistystiedoston lataamiseen asiakaslaitteelle PXE-käynnistyksen jälkeen. Asennuspisteeltä ladataan asiakaslaitteelle käynnistystiedosto, jonka avulla ladataan ja käynnistetään WinPE-käyttöjärjestelmä.

#### 2.3.4 WinPE

WinPE (Windows PE) on pieni käyttöjärjestelmä, jota käytetään käyttöjärjestelmien asentamiseen ja korjaamiseen. WinPE:n avulla voidaan:

- alustaa kiintolevy ennen käyttöjärjestelmän asentamista
- asentaa Windows tai muu käyttöjärjestelmä verkosta tai paikallisesta tietovarannosta
- suorittaa sovelluksia tai skriptejä verkosta tai paikallisesta tietovarannosta
- lisätä oma käyttöliittymä em. toimintojen suorittamiseksi
- automatisoida em. toimintoja. (Microsoft, 2018)

WinPE ei vaadi kiintolevyä toimiakseen. Käyttöjärjestelmä voidaan suorittaa kokonaisuudessaan muistista. WinPE-käyttöjärjestelmän perusversio vaatii 512Mt muistia. Käyttöjärjestelmän lisäksi käytettävät ajurit, sovellukset, skriptit yms. lisäävät muistintarvetta. (Microsoft, 2018)

Miradore-järjestelmällä tehtävä käyttöjärjestelmäasennus aloitetaan käynnistämällä asennuksen kohteena oleva laite WinPE-käyttöjärjestelmään. Tuettuja tapoja ovat:

- WinPE CD/DVD
- WinPE USB-asema
- WinPE verkkobootti
- WinPE kiintolevyltä. (Miradore, 2020, s. 791)

Tässä työssä tutkittiin ainoastaan verkosta käynnistettävää asennustapaa. WinPE ladataan verkosta työaseman muistiin tehtävälle RAM-levylle ja työasema käynnistetään RAM-levyltä.

WinPE-käyttöjärjestelmän avulla käynnistetään Miradoren oma asennusympäristö, jossa tehdään tarvittavat asetukset käyttöjärjestelmän asennuksiin liittyen.

### **2.3.5 Windows-levyjako**

Jaetut tiedostojärjestelmät mahdollistavat käyttäjille etätietokoneilla olevien tiedostojen avaamisen, lukemisen ja kirjoittamisen. Tämä mahdollistaa esimerkiksi useiden käyttäjien tarvitsemien tiedostojen tallentamisen kootusti yhteen paikkaan ja tiedostojen käyttämisen verkon yli. Miradore-järjestelmä käyttää levyjakoja asiakaslaitteen asennukseen tarvittavien tiedostojen jakamiseen. Työn tilaajan järjestelmässä Windows-levyjako toteutetaan asennuspisteen (Linux Debian) Samba-palvelulla.

Windows-levyjako toteutetaan CIFS (Common Internet Filesystem) -protokollan avulla. CIFS on julkaistu standardina, jota Microsoft edelleen kehittää, mutta se on huonosti dokumentoitu ja sisältää paljon salaista toimintaa. Kuitenkin, ryhmä kehittäjiä on onnistunut luomaan suositun ilmaisen ei-Windows -ympäristössä Windows-tiedostojakoa tarjoavan palvelun: Samban. (Adelstein & Lubanovic, 2007, ss. 164-165)

### **2.3.6 HTTP-tiedostojako**

HTTP (Hypertext Transfer Protocol) -tiedostojakoa käytetään Linux-asennuksissa pakettivarastona, joten asennuspisteillä olevaa dataa jaetaan myös HTTP-tiedostojakona. Tiedostojen jakaminen toteutetaan asennuspisteellä toimivan web-palvelimen avulla. Palvelimena käytetään Linuxin suosittua Apache-palvelinohjelmaa.

Palvelinohjelma ottaa vastaan asiakaslaitteiden lähettämiä HTTP-pyyntöjä TCP/IP -verkosta ja vastaa niihin. Palvelin voi palauttaa HTML-dokumentin, tekstitiedoston, kuvan tai yleensä minkälaisen tahansa tiedoston tai virheen. Linuxin paketinhallintaohjelmistot noutavat palvelimelta paketteja, jotka asennetaan päätelaitteelle.

### 2.3.7 Linux Debian -pakettivarasto (repository)

Debian-sovellukset järjestetään julkaisuksi pakettivarastoon. Julkaisu on saatavilla fyysisenä mediana (esim. CD/DVD) tai verkossa olevana pakettivarastona. Pakettien hallintasovellukset pystyvät asentamaan sovelluksia paikalliselta medialta tai verkosta. Pakettivarastoihin viitataan URI-osoitteella tiedostossa `"/etc/apt/sources.list"`. (Pollei, 2013, s. 55)

Tämän työn yhtenä tutkimuskohteena on oman pakettivaraston luominen työn tilaajan omille sovelluksille ja sen saattaminen asiakaslaitteiden käytettäväksi asennuksen aikana.

## 2.4 Asennusprosessi laitetta asennettaessa

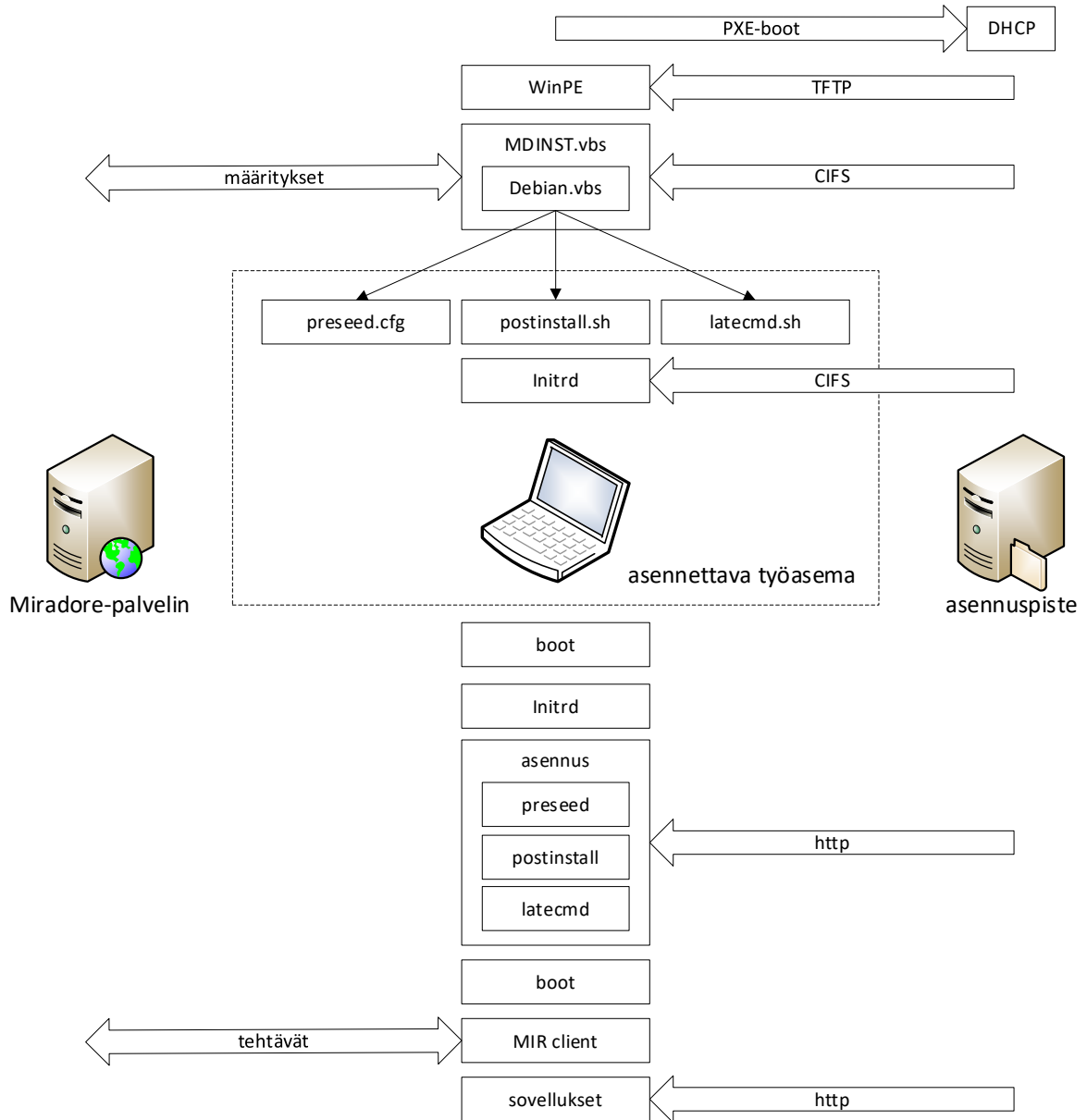
Laitteen (työasema, palvelin) asentaminen aloitetaan käynnistämällä laite verkosta (PXE). Verkosta (asennuspiste) ladataan WinPE-käyttöjärjestelmä ja käynnistetään järjestelmän toimittajan sovellus (MDInst.vbs), jonka avulla tehdään laitteen asennuksen vaatimat määritykset. Jos kyseessä on Linux Debian -laite, kopioidaan laitteelle asennuspisteeltä väliaikainen root-tiedostojärjestelmä (Initrd) ja suoritetaan Debian.vbs -skripti. Skriptin avulla luodaan asennuksessa käytettävä vastaustiedosto (preseed.cfg) sekä kaksi jälkiasennus-skriptiä (postinstall.sh, latecmd.sh), jotka suoritetaan asennuksen loppuvaiheessa. Debian.vbs -skriptin suorittaminen on toteutettu siten, että skripti sisällytetään (include) suorituksessa olevaan MDInst.vbs -skriptiin. Tästä johtuen Debian.vbs:n suorituksen aikana ovat käytettävissä kaikki MDInst.vbs -skriptin muuttujat. Linux Debian -laitteen asennusprosessi on kuvattu kuvassa 3.

Kun tarvittavat tiedostot on kopioitu ja luotu, käynnistetään laite uudestaan. Laite käynnistyy omalta massamuistiltaan väliaikaista root-tiedostojärjestelmää käyttäen ja aloittaa käyttöjärjestelmän asentamisen verkosta, käyttäen automaattisen asennuksen vastaustiedostossa (preseed.cfg) olevia määrityksiä.

Käyttöjärjestelmän asennuksen yhteydessä laitteelle asennetaan myös Miradore-asiakassovellus. Kun käyttöjärjestelmä on asennettu ja laite käynnistetty uudestaan, muodostaa asiakassovellus yhteyden Miradore-palvelimeen ja alkaa suorittamaan laitteelle

määritettyjä käyttöjärjestelmän asennuksen jälkeisiä toimenpiteitä. Yleensä nämä toimenpiteet ovat yksittäisten sovellusten asennuksia ja muita tarvittavia määrittämiä.

Kuva 3. Linux Debian -laitteen asennusprosessi.





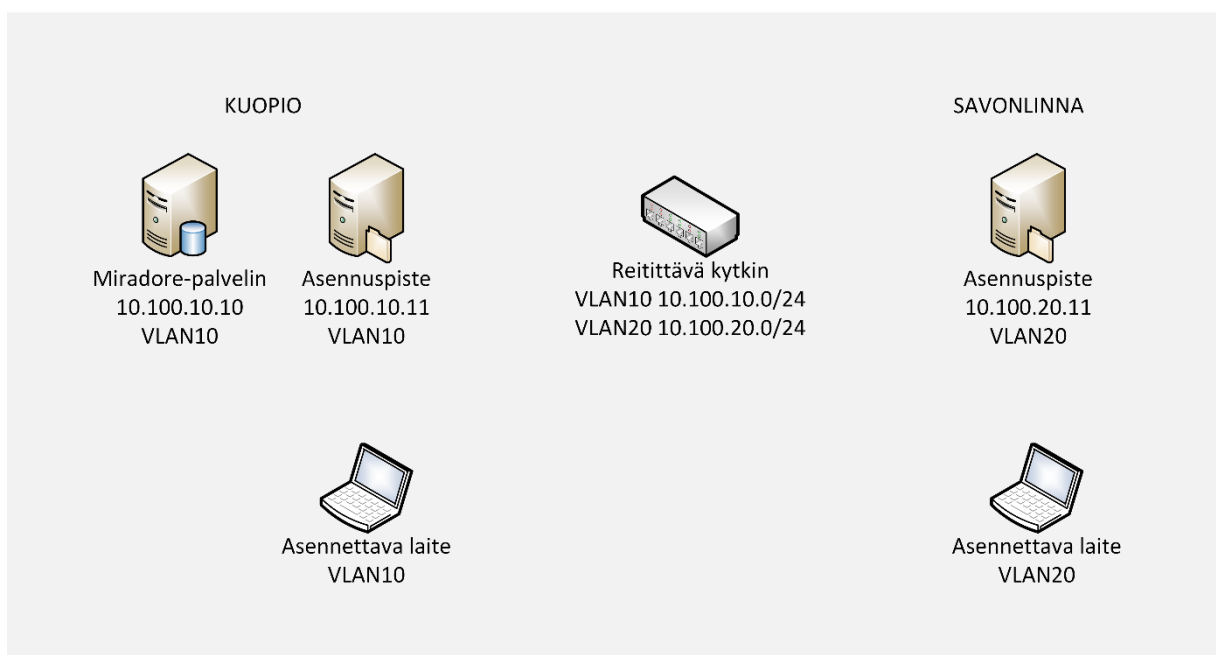
### 3 Kehittämistyön tavoite ja tarkoitus

Työn tavoitteena oli muokata Miradore-järjestelmää ja sen prosesseja niin, että se täyttää tilaajan vaatimukset Linux-laitteiden asennuksissa. Järjestelmän muokkaaminen tulisi toteuttaa niin, että järjestelmän päivittäminen, ylläpitäminen ja mahdollisten uusien ominaisuuksien toteuttaminen tuottaisi mahdollisimman vähän työtä. Lisäksi tavoitteena oli mahdollistaa myös se, että järjestelmän avulla pystytään asentamaan kolmannen osapuolen toimittamia kovennettuja Debian-käyttöjärjestelmiä.

### 4 Suunnittelu

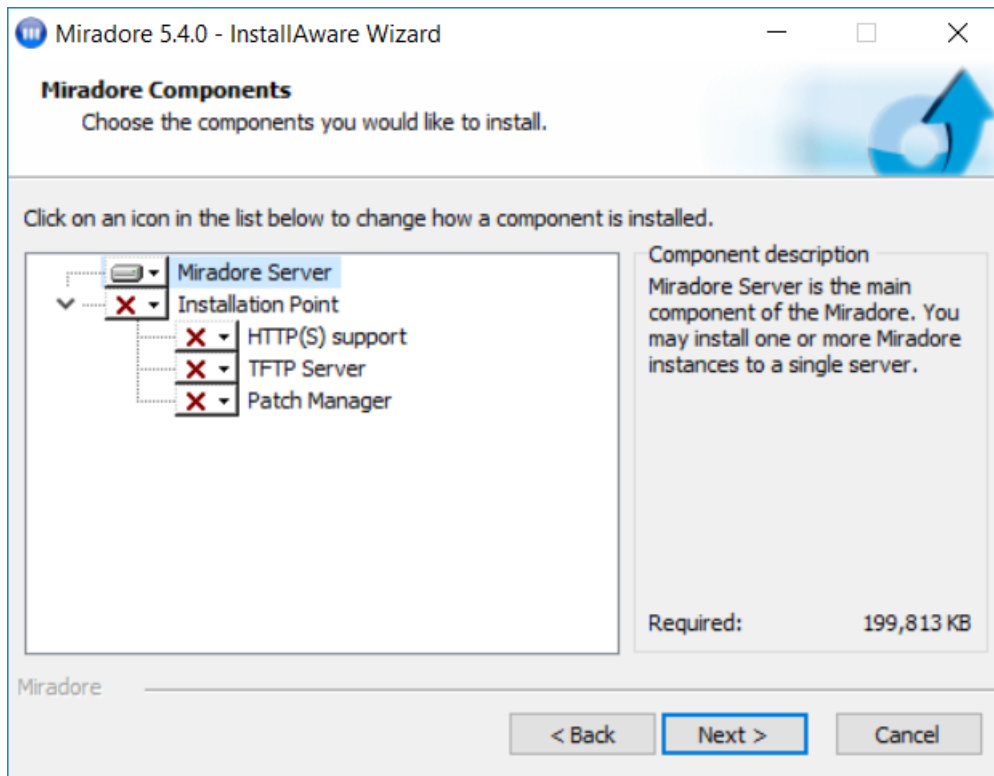
Suunnittelu ja toteutus aloitettiin asentamalla demo-ympäristö (Kuva 4) järjestelmästä. Demo-ympäristössä on kaksi lähiverkkoa ja niiden välinen reititys. Toisessa lähiverkossa (10.100.10.0/24) on Miradore-palvelin (.10) ja asennuspiste (.11). Toisessa lähiverkossa (10.100.20.0/24) on ainoastaan asennuspiste (.11). Palvelimet asennettiin virtuaalikoneiksi yhteen palvelinrataan (HP DL360 Gen9) ja reititys toteutettiin palvelimen ulkopuolella reitittävällä kytkimellä (HP FlexNetwork 5130 JG933A). Kytkimen portteihin voi myös liittää asennettavia laitteita molempiin lähiverkkoihin.

Kuva 4. Demo-ympäristö.



Miradore-palvelin asennettiin Windows Server 2016 -alustalle järjestelmän toimittajan asennusohjeiden mukaan. Oletusasetuksista poikettiin sen verran, että asennuspiste-toiminnallisuutta ei asennettu itse Miradore-palvelimen yhteyteen (Kuva 5).

Kuva 5. Miradore-palvelimen asennusvalinnat.



Asennuspisteet asennettiin järjestelmän toimittajan ohjeiden mukaan erillisinä Debian-palvelimina. Asennuspisteiltä jaettavalle datalle luotiin hakemisto `/var/miradore/setup/`. Tähän sijaintiin viitataan myöhemmin asennushakemistona. Hakemisto jaetaan verkkoon CIFS-levyjakona (Samba) sekä HTTP-palveluna (Apache 2). Asennuspisteelle luotiin Samballe kaksi käyttäjää levyjaon käyttämistä varten ja näille käyttäjille määritettiin vastaavat luku- ja kirjoitusoikeudet Samban konfiguraatioon:

- mir-reader (lukuoikeudet)
- mir-writer (kirjoitusoikeudet)

Miradore-järjestelmä käyttää näitä käyttäjiä asennusten yhteydessä ja datan automaattiseen replikointiin asennuspisteiden välillä. Käyttäjien avulla voidaan myös yhdistää työasema asennuspisteen levyjakoon ylläpitotoimenpiteitä varten.

Asennushakemistoon kopioitiin tarvittava data Miradore-palvelimelta ja tuotiin Debian 10 -käyttöjärjestelmä järjestelmän toimittajan ohjeiden mukaan.

Järjestelmään luotiin kuvitteellinen organisaatio (Yritys XYZ), jolla on kaksi toimipistettä eri sijainneissa (Kuopio ja Savonlinna). Toimipisteet sidottiin kahteen edellä mainittuun lähiverkkoon. Kahta toimipistettä ja lähiverkkoa käyttämällä voimme todentaa, että muokattu järjestelmä on laajennettavissa tilaajan asennusympäristöön, jossa on kymmeniä toimipisteitä ja verkkoja.

Järjestelmän toimintaa tutkittiin suorittamalla asennuksia demojärjestelmällä. Asennusten yhteydessä tutkittiin Debian.vbs -tiedoston suorituksen aikaisia muuttujia ja tallennettiin alkuperäinen Debian.vbs -tiedosto, sekä järjestelmän oletusasetuksilla asennettavalle laitteelle luodut tiedostot:

- Debian.vbs (Liite 1)
- preseed.cfg (Liite 2)
- latecmd.sh (Liite 3)
- postinst.sh (Liite 4)

Asennusten yhteydessä huomattiin, että Miradoren asiakassovellus (client) ei asentunut asennettaville laitteille. Tarkempi selvittely osoitti, että Debian 10 -käyttöjärjestelmä ei hyväksynyt pakettivarastoa (repository), josta asiakassovellus asennettaisiin. Järjestelmän perusasetuksilla asiakassovellus asennetaan asennushakemiston \_System-hakemiston alla olevasta valmiista pakettivarastosta. Tämä pakettivarasto on luotu vanhemmille Debian -versioille, eikä Debian 10 enää hyväksy vastaavaa rakennetta. Kerroimme löydöstämme järjestelmän toimittajalle, mutta emme muuten huolestuneet tilanteesta, koska asiakassovellus tullaan myöhemmin siirtämään tilaajan omaan pakettivarastoon.

Työtä suunnitellessa tuli selväksi, että työn tavoitteisiin on mahdollista päästä monella eri tavalla. Parhaiten tavoitteet täyttyivät kuitenkin seuraavilla perusteilla:

#### **4.1 Järjestelmästä muokataan ainoastaan Debian.vbs –tiedostoa**

Debian.vbs -skripti on keskeisessä asemassa asennuksen alkuvaiheessa. Sen suorittaminen luo myöhemmin asennuksessa käytettävät tiedostot (preseed.cfg, postinstall.sh ja latecmd.sh). Vain yhden tiedoston muokkaaminen helpottaa myöhemmin järjestelmän ylläpitoa ja päivittämistä.

#### **4.2 Asennuksen vastaustiedostosta luodaan pohjatiedosto**

Tilaajan vaatimuksena oli, että järjestelmästä pystyy asentamaan useita eri käyttöjärjestelmäkoonpanoja. Tämä aiheuttaa sen, että käytettävissä pitää olla useita erilaisia automaattisen asennuksen vastaustiedostoja (preseed.cfg), riippuen loppukäyttäjän asennuksen aikana tekemistä valinnoista. Oletusasetuksilla vastaustiedosto luodaan kokonaan Debian.vbs -skriptin suorituksen yhteydessä suoraan asennuksen kohteena olevalle laitteelle.

Useiden loppukäyttäjän valintojen mukaisten vastaustiedostojen luominen olisi mahdollista toteuttaa Debian.vbs -skriptin avulla, mutta skriptistä tulisi monimutkainen ja vaikeasti ylläpidettävä. Järjestelmän pitää myös mahdollistaa kolmannen osapuolen toimittamien käyttöjärjestelmäkoonpanojen käyttäminen, eikä tällä kolmannella osapuolella ole pääsyä tilaajan järjestelmään.

Tilaajan vaatimukset täyttävä toteutus saadaan aikaiseksi mahdollistamalla vastaustiedostojen pohjatiedoston (template) käyttäminen. Pohjatiedostot tallennetaan asennuspisteelle. Debian.vbs -skriptin suorituksen yhteydessä valitaan ja noudetaan asennuspisteeltä loppukäyttäjän valintoihin soveltuva pohjatiedosto ja sitä muokataan tarvittavilta osin ennen asennettavalle laitteelle kopioimista.

#### **4.3 Jälkiasennus-skripteistä luodaan pohjatiedostot**

Kolmannen osapuolen toteuttamat käyttöjärjestelmän kovennukset vaativat mahdollisesti toimintoja, joita ei pystytä toteuttamaan asennuksen vastaustiedoston avulla. Tämä mahdollistetaan luomalla myös postinst.sh -skriptistä pohjatiedosto, joka voidaan

tarvittaessa korvata kolmannen osapuolen toimittamalla tiedostolla. Tätä tiedostoa ei muokata tilaajan järjestelmässä, eikä sen avulla toteuteta tilaajan omia asennuksen aikaisia toimenpiteitä.

Myös toisesta asennuksen loppuvaiheessa suoritettavasta skriptistä (latecmd.sh) luodaan pohjatiedosto. Pohjatiedostoa muokataan Debian.vbs -skriptin suorittamisen yhteydessä ja sen avulla toteutetaan tilaajan omia asennukseen liittyviä toimintoja, kuten Miradoren asiakasohjelman asentaminen ja käyttäjätilien luominen.

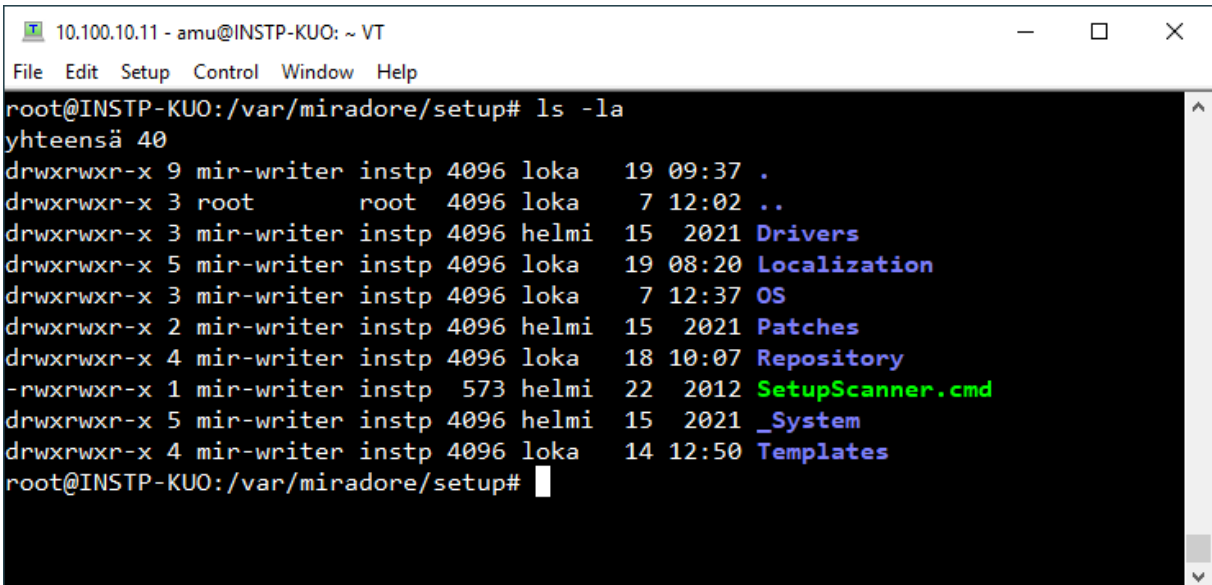
## 5 Toteutus

Järjestelmän muokkaus toteutettiin vaiheittain ja näiden vaiheiden aikana Debian.vbs -skripti kirjoitettiin kokonaan uudestaan.

### 5.1 Pohjatiedostojen luominen

Ensimmäisessä vaiheessa luotiin asennuksen tarvitsemat pohjatiedostot asennuspisteelle ja toteutettiin niiden käyttäminen Debian.vbs -skriptiin. Pohjatiedostoina käytettiin aikaisemmin tallennettuja alkuperäisiä tiedostoja. Pohjatiedostoja varten luotiin Templates-hakemisto asennushakemiston juureen (kuva 6).

Kuva 6. Asennushakemiston sisältö asennuspisteellä.



```

10.100.10.11 - amu@INSTP-KUO: ~ VT
File Edit Setup Control Window Help
root@INSTP-KUO:/var/miradore/setup# ls -la
yhteensä 40
drwxrwxr-x 9 mir-writer instp 4096 loka  19 09:37 .
drwxrwxr-x 3 root          root  4096 loka  7 12:02 ..
drwxrwxr-x 3 mir-writer instp 4096 helmi 15 2021 Drivers
drwxrwxr-x 5 mir-writer instp 4096 loka  19 08:20 Localization
drwxrwxr-x 3 mir-writer instp 4096 loka  7 12:37 OS
drwxrwxr-x 2 mir-writer instp 4096 helmi 15 2021 Patches
drwxrwxr-x 4 mir-writer instp 4096 loka  18 10:07 Repository
-rwxrwxr-x 1 mir-writer instp  573 helmi 22 2012 SetupScanner.cmd
drwxrwxr-x 5 mir-writer instp 4096 helmi 15 2021 _System
drwxrwxr-x 4 mir-writer instp 4096 loka  14 12:50 Templates
root@INSTP-KUO:/var/miradore/setup#

```

Samassa yhteydessä luotiin valmiiksi myös Repository ja Localization -hakemistot myöhempää käyttöä varten.

Tilajan vaatimuksena on, että järjestelmän avulla voidaan asentaa useita erilaisia käyttöjärjestelmäkoonpanoja. Tämä aiheuttaa sen, että käytettävissä pitää olla useita pohjatiedostoja ja pohjatiedostojen valinta tehdään loppukäyttäjän järjestelmässä tekemien valintojen mukaan. Tämä toiminnallisuus toteutettiin niin, että käyttöön otettavat pohjatiedostot valitaan järjestelmään asetettujen laiteroolien (Device role) mukaan.

Templates-hakemistoon luotiin laiteroolin (DEBIAN) mukainen hakemisto pohjatiedoille ja pohjatiedostot sijoitettiin kyseiseen hakemistoon. Esimerkki laiteroolista kuvassa 7.

Kuva 7. Laiterooli (DEBIAN) Miradore-järjestelmässä.

## Yritys XYZ > Device role

DEBIAN

Created 16.2.2021 | Modified 16.2.2021

General <span>(i)</span>	
Name:	DEBIAN
Short name:	DEB
Description:	GNOME -työpöytä
Role defaults <span>(i)</span>	
Automated installation:	Yes
Default operating system:	<a href="#">Debian 10 64-bit</a>
Additional information <span>(i)</span>	
Attachments:	-
Comments:	-

Järjestelmään luotiin myös toinen laiterooli (DEBIAN SERVER), että voidaan myöhemmin testata useiden erilaisten laitekoonpanojen asentamista.

Debian.vbs -skriptin kirjoittaminen aloitettiin kokonaan alusta (Ohjelmakoodi 1). Tässä vaiheessa otettiin ainoastaan käyttöön aikaisemmin tallennetut pohjatiedostot laiteroolia vastaavasta hakemistosta. Laiteroolin nimi on käytettävissä muuttujana (DeviceRole) Debian.vbs -skriptissä. Laiteroolien nimissä voidaan käyttää välilyöntejä ja tiedostojärjestelmissä välilyönnit hakemistojen tai tiedostojen nimissä aiheuttavat usein ongelmia. Ongelmien välttämiseksi päätettiin korvata laiteroolin nimessä olevat mahdolliset välilyönnit väliviivalla ja tallentaa pohjatiedostot näitä vastaaviin hakemistoihin, esim. DEBIAN-SERVER.

## Ohjelmakoodi 1. Debian.vbs ensimmäisessä vaiheessa.

```
' /var/miradore/setup/_System/_Start/Debian.vbs
' Ari Mustonen, INTIM19A6

ShowProgress "WriteLine", ""

Dim TemplateName, PreseedTemplate, LateTemplate, PostTemplate

' Korvataan laiteroolin mahdollinen välilyönti väliviivalla
TemplateName = Replace(DeviceRole, " ", "-")

' Määritetään pohjatiedostojen sijainti laitteen roolin (DEVICE ROLE) mukaan
PreseedTemplate = SetupDrive & "\\Templates\\" & TemplateName & "\\preseed.cfg"
LateTemplate = SetupDrive & "\\Templates\\" & TemplateName & "\\latecmd.sh"
PostTemplate = SetupDrive & "\\Templates\\" & TemplateName & "\\postinst.sh"

' Luetaan preseed.cfg templaatti muistiin
Dim strContent
ShowProgress "Write", "- Reading preseed.cfg template... "
Set fPreseed = g_objFSO.OpenTextFile(PreseedTemplate)
strContent = fPreseed.ReadAll
fPreseed.Close
ShowProgress "WriteLine", "Done"

' Kirjoitetaan preseed MDInst.vbs käyttämään muuttujaan (fUna)
fUna.Write(strContent)

' Kopioidaan latecmd.sh asennuspisteeltä asennettavalle laitteelle
ShowProgress "Write", "- Reading latecmd.sh template... "
g_objFSO.CopyFile LateTemplate, strTargetDir & "\\latecmd.sh"
ShowProgress "WriteLine", "Done"

' Kopioidaan postinstall.sh asennuspisteeltä asennettavalle laitteelle
ShowProgress "Write", "- Reading postinstall.sh template... "
g_objFSO.CopyFile PostTemplate, strTargetDir & "\\postinst.sh"
ShowProgress "WriteLine", "Done"
```

Automaattisen asennuksen vastaustiedoston (preseed.cfg) pohjatiedostoa ei voitu kopioida suoraan asennettavalle laitteelle. Järjestelmän rakenteesta johtuen automaattisen asennuksen (unattended install) vastaustiedosto on tässä vaiheessa vielä MDInst.vbs -skriptissä avatussa avoimessa tiedostomuuttujassa (fUna) ja lopullinen tiedosto tallennetaan asennettavalle laitteelle vasta Debian.vbs -skriptin suorittamisen jälkeen. Tämä ratkaistiin siten, että käyttämämme pohjatiedosto luettiin asennuspisteeltä ja ainoastaan kopioitiin



kyseiseen muuttujaan. Muut pohjatiedostot (latecmd.sh ja postinst.sh) kopiottiin sellaisenaan asennettavalle laitteelle.

Asennuksia testattiin molemmilla luoduilla laiterooleilla ja asennukset onnistuivat moitteitta. Toki tässä vaiheessa myös laitteista tuli samanlaisia, koska molemmilla laiterooleilla käytettiin samoja pohjatiedostoja, luettuina eri hakemistoista.

## 5.2 Oman pakettivaraston (repository) lisääminen

Toisessa vaiheessa toteutettiin tilaajan oman pakettivaraston lisääminen järjestelmään. Järjestelmä jakaa perusasetuksillaan HTTP-palveluna koko asennushakemiston sisällön. Tämä on tietoturvan kannalta huono ratkaisu, koska asennuspisteellä on huomattava määrä dataa, jota ei tarvita Linux-asennuksissa, eikä tähän dataan täten ole tarvetta avata pääsyä.

Tehtiin päätös, että HTTP-palveluna jaetaan ainoastaan Linux-asennuksissa tarvittava data. Asennuspisteen Apache2 web-palvelinohjelmisto konfiguroitiin (Ohjelmakoodi 2) niin, että asennuspisteeltä jaetaan HTTP-jakona ainoastaan käyttöjärjestelmän sijainti sekä tilaajan omille sovelluksille luotavan pakettivaraston sijainti:

- /var/miradore/setup/OS/Linux/Debian (käyttöjärjestelmä)
- /var/miradore/setup/Repository (tilaajan oma pakettivarasto)

Ohjelmakoodi 2. Web-palvelinohjelmiston asetustiedosto.

```
# Apache2 - /etc/apache2/sites-available/instp.conf
# Ari Mustonen, INTIM19A6

Alias /Debian /var/miradore/setup/OS/Linux/Debian
Alias /Repository /var/miradore/setup/Repository

<Directory /var/miradore/setup/OS/Linux/Debian>
    Options Indexes FollowSymlinks
    IndexOptions FancyIndexing
    Require all granted
</Directory>

<Directory /var/miradore/setup/Repository>
    Options Indexes FollowSymlinks
    IndexOptions FancyIndexing
```

```
Require all granted
</Directory>
```

Asennushakemiston Repository-hakemistoon luotiin pool-hakemisto ja sen alle komponenttihakemisto main. Main-komponentin alle luotiin vielä hakemisto Miradoren asiakasohjelmalle ja kopioitiin asiakasohjelman asennuspaketti kyseiseen hakemistoon:

```
/var/miradore/setup/Repository/pool/main/miradore-
client/miradoreclient_1.6.6_amd64.deb
```

Jotta Debianin paketinhallintasovellukset pystyisivät käyttämään pakettivarastoa, pitää pakettivarasto julkaista (release) ja allekirjoittaa (sign) julkaisutiedostot OpenPGP-allekirjoitusavaimella. (Debian, n.d.)

Julkaisutiedostojen allekirjoittamista varten asennuspisteelle luotiin salainen OpenPGP-avain ja tälle avaimelle erillinen allekirjoitusavain. Avaimen julkinen osa vietiin ASCII-tiedostoon (firmaxyz.public.key) ja tiedosto kopioitiin oman pakettivarastomme juureen (/var/miradore/setup/Repository), josta se on käytettävissä päätelaitteita asennettaessa. Avaimien luominen ja julkisen avaimen tallentaminen tiedostoksi on kuvattu liitteessä 5.

Pakettivaraston julkaisua ja allekirjoittamista varten kirjoitettiin kaksi shell-skriptiä /usr/local/bin -hakemistoon ja skripteille määritettiin suoritusoikeudet root-käyttäjälle:

- mir-release-repository (Liite 6)
- mir-sign-release (Liite 7)

Skriptit toteutettiin siten, että ne suoritetaan oman pakettivarastomme juuressa.

Ensimmäinen skripti luo tarvittavat julkaisutiedostot pakettivaraston dists-hakemistoon.

Toisella skriptillä allekirjoitetaan julkaisutiedostot aikaisemmin luodulla avaimella.

Allekirjoitus-skriptille annetaan parametrina käytettävän avaimen tunnus. Skripti kysyy avaimen salasanaa, varmistaa salasanan toimivuuden ja jos salasana on oikea, allekirjoittaa julkaisutiedostot. Esimerkki pakettivaraston julkaisemisesta ja allekirjoittamisesta kuvassa 8.

Kuva 8. Pakettivaraston julkaiseminen ja allekirjoittaminen.

```

10.100.10.11 - amu@INSTP-KUO: ~ VT
File Edit Setup Control Window Help
root@INSTP-KUO: /var/miradore/setup/Repository# mir-release-repository
Skannataan paketit komponentille main ja arkkitehtuurille amd64
dpkg-scanpackages: info: Wrote 1 entries to output Packages file.
Luodaan Release-tiedosto
Release-tiedosto valmis
MUISTA ALLEKIRJOITTA REPOSITORY!

root@INSTP-KUO: /var/miradore/setup/Repository# mir-sign-release ari@firmaxyz.x
Avaimen salasana (ei näy kirjoitettaessa):

Salasana OK. Jatketaan...

Luodaan Release.gpg...
gpg: using subkey 3BB6B203CAC42931 instead of primary key CDCFBCC93872BC8C
gpg: writing to './dists/xyz/Release.gpg'
gpg: RSA/SHA256 signature from: "3BB6B203CAC42931 Ari Mustonen (Repositoryn allekirjoitusavain) <ari@firmaxyz.x>"

Luodaan InRelease...
gpg: using subkey 3BB6B203CAC42931 instead of primary key CDCFBCC93872BC8C
gpg: writing to './dists/xyz/InRelease'
gpg: RSA/SHA256 signature from: "3BB6B203CAC42931 Ari Mustonen (Repositoryn allekirjoitusavain) <ari@firmaxyz.x>"

OK

root@INSTP-KUO: /var/miradore/setup/Repository#

```

Oman pakettivaraston valmistelujen jälkeen muokattiin automaattisen asennuksen vastaustiedoston Repository-osuus (Ohjelmakoodi 3) vastaamaan tekemiämme muutoksia.

Ohjelmakoodi 3. Preseed.cfg –pohjatiedoston Repository-osuus toisessa vaiheessa.

```

#
# Repository
#
d-i mirror/country string manual
d-i mirror/protocol select http
d-i mirror/http/hostname string 10.100.10.11:80
d-i mirror/http/directory string /Debian/10
d-i mirror/http/proxy string
d-i mirror/suite string buster
d-i mirror/udeb/suite string buster
d-i apt-setup/local0/repository string deb http://10.100.10.11:80/Repository xyz main
d-i apt-setup/local0/source boolean false
d-i apt-setup/local0/key string http://10.100.10.11:80/Repository/firmaxyz.public.key
d-i debian-installer/allow_unauthenticated boolean false
d-i apt-setup/services-select select none

```

Samassa yhteydessä pohjatiedostosta poistettiin Debianin security-pakettivarasto (security.debian.org), koska tilaajan verkosta ei ole pääsyä internetiin.

Muutosten toiminta testattiin koeasennuksilla. Asennukset onnistuivat moitteitta ja myös Miradoren asiakasohjelmisto asentui omasta pakettivarastostamme.

### 5.3 Jälkiasennus-skriptien muokkaaminen

Kolmannessa vaiheessa käsittelyyn otettiin jälkiasennus-skriptit. Suunnitelmamme mukaan toinen skripteistä (postinst.sh) jätettiin kolmannen osapuolen tarvitsemille toiminteille ja tilaajan omat toiminnot keskitettiin yhteen (latecmd.sh) skriptiin, joka suoritetaan asennuksen loppuvaiheessa.

Molemmista skripteistä tehtiin uudet pohjatiedostot. Latecmd.sh -skriptin pohjatiedostoon (Ohjelmakoodi 4) ohjelmoitiin toiminallisuus, joka lisää asennettavalle laitteelle kaksi eritasoista käyttäjätunnusta. Näin käyttäjätunnusten hallinta saatiin tilaajan vaatimusten mukaisesti piiloon loppukäyttäjiltä. Näitä käyttäjätunnuksia pystytään tarvittaessa muuttamaan laiteroolia vastaavaa pohjatiedostoa muokkaamalla.

Ohjelmakoodi 4. Latecmd.sh -pohjatiedosto kolmannessa vaiheessa.

```
# latecmd.sh -pohjatiedosto
# Ari Mustonen, INTIM19A6

#!/bin/sh
echo "`date` === latecmd script started ===" >> /var/log/latecmd.log

# Luodaan käyttäjätilit ja määritetään salasanat
useradd -m -s /bin/bash user
useradd -m -s /bin/bash admin
usermod -aG sudo admin

echo -E 'user:user-salanasana' | chpasswd
echo -E 'admin:admin-salanasana' | chpasswd

# Skriptiä täydennetään Debian.vbs:n suorituksen aikana
```

Debian.vbs -skriptiä muokattiin (Ohjelmakoodi 5) siten, että latecmd.sh -skriptin pohjatiedostoon lisätään Miradore-asiakasohjelman asentamisen vaatimat toimenpiteet. Tämä toiminnallisuus kopioitiin melko suoraan järjestelmän alkuperäisestä Debian.vbs -skriptistä (Liite 1). Viimeisenä tiedostoon kirjoitetaan rivi, joka suorittaa toisen (postinst.sh) jälkiasennus-skriptin (Liite 12).

Ohjelmakoodi 5. Latecmd.sh -pohjatiedoston käsittely (Debian.vbs) kolmannessa vaiheessa.

```
' Luetaan latecmd.sh -pohjatiedosto muistiin
ShowProgress "Write", "- Reading latecmd.sh template... "
Set fLateTemplate = g_objFSO.OpenTextFile(LateTemplate)
strContent = fLateTemplate.ReadAll
fLateTemplate.Close
ShowProgress "WriteLine", "Done"

' Luodaan uusi latecmd.sh laitteelle
ShowProgress "Write", "- Editing latecmd.sh... "
Set fLateFile = g_objFSO.CreateTextFile(strTargetDir & "\latecmd.sh", True)

' Kirjoitetaan pohjatiedoston sisältö tiedostoon
fLateFile.Write(strContent)

' Luodaan Miradore clientin konffifile
WriteLinuxFile fLateFile, "echo # Miradore client configuration > /etc/miradore.conf"
Dim strClientParm
For Each strClientParm in Split(ClientParams, vbCrLf)
    WriteLinuxFile fLateFile, "echo " & strClientParm & " >> /etc/miradore.conf"
Next

' Miradore clientin asennus
WriteLinuxFile fLateFile, "apt-get install --yes miradoreclient >> /var/log/latecmd.log"

' Postinst.sh suorittaminen
WriteLinuxFile fLateFile, "bash /tmp/postinst.sh"
WriteLinuxFile fLateFile, "exit 0"

' Suljetaan latecmd.sh
fLateFile.Close
ShowProgress "WriteLine", "Done"
```

Automaattisen asennuksen vastaustiedoston Post-commands -osiota muokattiin (Ohjelmakoodi 6) vastaamaan tehtyjä muutoksia. Myös laitteen nimen määrittäminen siirrettiin (Ohjelmakoodi 7) latecmd.sh -skriptistä automaattisen asennuksen vastaustiedoston Network config -osiioon.

Ohjelmakoodi 6. Automaattisen asennuksen vastaustiedoston Post-commands -osio.

```
#
# Post-commands
#
d-i preseed/late_command string cp /postinst.sh /target/tmp/postinst.sh; cp /latecmd.sh
/target/tmp/latecmd.sh; chroot /target sh -c "/bin/bash /tmp/latecmd.sh"
#
```

Ohjelmakoodi 7. Automaattisen asennuksen vastaustiedoston Network config -osio

kolmannessa vaiheessa.

```
#
# Network config
#
d-i netcfg/choose_interface select auto
d-i netcfg/get_hostname string L0000005
d-i netcfg/link_wait_timeout string 10
d-i netcfg/dhcp_timeout string 60
d-i netcfg/dhcpv6_timeout string 60
d-i netcfg/wireless_wep string
#
```

## 5.4 Lokalisointi

Asennettaville laitteille määritetään laiteroolin lisäksi pakollisena tietona myös laitteen sijainti. Lokalisoinnilla pyrittiin mahdollistamaan laitteiden asetusten määrittäminen sijaintien mukaan. Tavoitteena oli mahdollistaa eri salasanojen käyttäminen eri sijainneissa.

Järjestelmään luotaville sijainneille (Kuva 9) määritetään lyhytnimi (Short name). Lyhytnimi on käytettävissä muuttujassa (LocSN) Debian.vbs -skriptin suorituksen aikana. Lokalisointi toteutettiin siten, että asennushakemiston Localization-hakemistoon luotiin sijaintien lyhytnimiä vastaavat hakemistot ja hakemistoihin luotiin laiteroolien nimiä vastaavat tiedostot, joiden sisältönä on salattuna salasanat perus- ja ylläpitokäyttäjille.

Kuva 9. Järjestelmään määritetty sijainti (Location).

## Yritys XYZ > Location

Savonlinna

Created 15.2.2021 | Modified 7.10.2021

General <span>(i)</span>	
Name:	Savonlinna
Short name:	SLN
Status:	Active
Parent:	<a href="#">Yritys XYZ</a>
Responsible:	
Description:	
Location selectable for:	<input checked="" type="checkbox"/> Asset <input checked="" type="checkbox"/> User

Salasanatiedostojen luomista varten ohjelmoitiin shell-skripti (Liite 8). Skripti kysyy kaksi salasanaa ja kirjoittaa ne salattuina passwords.pwd -tiedostoon. Tiedosto nimetään laiteroolin mukaiseksi ja kopioidaan haluttuun Localization-hakemiston alihakemistoon. Esimerkki skriptin käyttämisestä kuvassa 10. Localization-hakemistoon luotiin myös DEFAULT-hakemisto ja sinne laiterooleja vastaavat salasanatiedostot. Jos sijaintia ja laiteroolia vastaavaa salasanatiedostoa ei löydetä, niin tiedostoa haetaan DEFAULT-hakemistosta.

Kuva 10. Salasanatiedoston luominen ja nimeäminen.

```

10.100.10.11 - amu@INSTP-KUO: ~ VT
File Edit Setup Control Window Help
root@INSTP-KUO: /var/miradore/setup/Localization/SLN# mir-create-pwdfile

Scripti kysyy salasanat ja kirjoittaa ne salattuina passwords.pwd -tiedostoon
Vaihda tiedoston nimi vastaamaan laiteroolia

peruskäyttäjän salasana:
user-password

ylläpitokäyttäjän salasana:
admin-password

root@INSTP-KUO: /var/miradore/setup/Localization/SLN# mv passwords.pwd DEBIAN-SERVER.pwd
root@INSTP-KUO: /var/miradore/setup/Localization/SLN# cat DEBIAN-SERVER.pwd
$6$o98Y6dmgXQMW$EKfPZhn5TCXmRczfGpUy31b85kwZWrPFvfvf7483xvX2pCogGGimmUYbgZ7y1pHmNdDHBdkOUFGJa.KHha01v
d.
$6$2jULDPvHbVpdVz$Cu5cUCA5ZaCayh10deOVHG1s0AwnAoCE6GkHujS.XbZDi5SKNIZK3fn329qNnVy3QoKMYTheZpN1Z4w6ny
bAC.

Tiedosto luotu 'mir-create-pwdfile' scriptillä.
ÄLÄ MUOKKAA TIEDOSTOA MUILLA TAVOILLA!
root@INSTP-KUO: /var/miradore/setup/Localization/SLN#

```

Latecmd.sh -pohjatiedostoa muokattiin (Ohjelmakoodi 8) niin, että käyttäjätilien luomisessa käytetään salattuja salasanonoja. Pohjatiedostoon korvattiin salasanat muuttujilla (`_USERPWD_` ja `_ADMINPWD_`), joiden sisältö korvataan Debian.vbs -skriptissä salasanatiedostosta luettavilla salasanonoilla (Ohjelmakoodi 9) skriptiin lisättyä funktiota (Ohjelmakoodi 10) käyttäen.

Ohjelmakoodi 8. Käyttäjätilien luominen latecmd.sh -skriptissä.

```

# Luodaan käyttäjätilit ja määritetään salasanat
useradd -m -p '_USERPWD_' -s /bin/bash user
usermod -aG cdrom,audio,video,netdev,bluetooth user
useradd -m -p '_ADMINPWD_' -s /bin/bash admin
usermod -aG sudo,netdev admin

```



## Ohjelmakoodi 9. Salasanatiedostojen käsittely Debian.vbs -skriptissä.

```
' Luetaan salasanat lokalisoititiedostosta
PwdFileName = TemplateName & ".pwd"
If g_objFSO.FileExists(SetupDrive & "\Localization\" & LocSN & "\" & PwdFileName) Then
    PwdFile = SetupDrive & "\Localization\" & LocSN & "\" & PwdFileName
    ShowProgress "WriteLine", "- Localization file found for location (" & LocSN & ")."
ElseIf g_objFSO.FileExists(SetupDrive & "\Localization\DEFAULT\" & PwdFileName) Then
    PwdFile = SetupDrive & "\Localization\DEFAULT\" & PwdFileName
    ShowProgress "WriteLine", "- Localization file not found. Using DEFAULT."
Else
    ShowProgress "WriteLine", "- ERROR! Localization file not found!"
End If

Set fPwd = g_objFSO.OpenTextFile(PwdFile)
uPwd = fPwd.ReadLine
aPwd = fPwd.ReadLine
fPwd.Close

' Kirjoitetaan (korvataan) salasanat muistissa olevaan pohjatiedostoon
ReplaceVariable strContent, "_USERPWD_", uPwd
ReplaceVariable strContent, "_ADMINPWD_", aPwd
```

## Ohjelmakoodi 10. Pohjatiedostojen muuttujia käsittelevä funktio.

```
' Funktio jolla muokataan pohjatiedostoja
Sub ReplaceVariable(strContent, template, value)
    strContent = Replace(strContent, template, value, 1, -1, 1)
End Sub
```

### 5.5 Automaattisen asennuksen vastaustiedoston käsittely

Automaattisen asennuksen vastaustiedostossa määritetään asetuksia, joita on järjestelmän oikein toimimiseksi tarve muuttaa vastaamaan järjestelmässä tehtyjä valintoja:

- Käytettävän asennuspisteen IP-osoite
- Laitteen nimi
- Root-käyttäjän salasana

Vastaustiedoston muokkaaminen toteutettiin siten, että muutettavat tiedot korvattiin pohjatiedostossa muuttujilla, joiden sisältö muutetaan Debian.vbs -skriptissä vastaamaan loppukäyttäjän tekemiä valintoja (Ohjelmakoodi 11). Muuttujien korvaamiseen käytettiin

samaa funktiota kuin `latecmd.sh` -pohjatiedoston muokkaamiseen. Tarvittavat tiedot muuttujien korvaamiseen ovat käytettävissä muuttujissa (`strHttpInstPointServer`, `strCompName` ja `AdminPassword`) Debian.vbs -skriptin suorituksen aikana.

Ohjelmakoodi 11. Preseed -pohjatiedoston muuttujien käsittely.

```
' Muokataan preseed -pohjatiedoston muuttujat asennusvalintojen mukaisiksi
ShowProgress "Write", "- Replacing template variables... "
ReplaceVariable strContent, "_HTTPINSTPOINTSERVER_", strHttpInstPointServer
ReplaceVariable strContent, "_INSTALLERPWD_", AdminPassword
ReplaceVariable strContent, "_COMPNAME_", strCompName
ShowProgress "WriteLine", "Done"

' Kirjoitetaan preseed MDInst.vbs käyttämään muuttujaan (fUna)
fUna.Write(strContent)
```

Lisäksi automaattisen asennuksen vastaustiedostoa muokattiin (Ohjelmakoodi 12) niin, että root-käyttäjä korvattiin installer-käyttäjällä. Tämä muutos lisää asennettujen laitteiden tietoturvaa, koska laitteisiin ei pysty muutoksen jälkeen kirjautumaan root-käyttäjänä. Lopullinen valmis automaattisen asennuksen vastaustiedosto liitteessä 10.

Ohjelmakoodi 12. Root-käyttäjän korvaaminen installer-käyttäjällä.

```
#
# Root user
#
d-i passwd/root-login boolean false
#
# Normal user
#
d-i passwd/make-user boolean false
d-i passwd/user-fullname string installer
d-i passwd/username string installer
d-i passwd/user-password password _INSTALLERPWD_
d-i passwd/user-password-again password _INSTALLERPWD_
#
```

Järjestelmään luotiin lopuksi tarvittavat pohjatiedostot molemmille (DEBIAN ja DEBIAN SERVER) laiteroleille ja lokalisointitiedostot molemmille sijainneille. Lopullinen Debian.vbs -skripti ja DEBIAN-roolin pohjatiedostot liitteinä:

- Debian.vbs (Liite 9)
- preseed.cfg (Liite 10)
- latecmd.sh (Liite 11)
- postinst.sh (Liite 12)

DEBIAN SERVER -laiteroolin pohjatiedostoina käytettiin samoja pohjatiedostoja, mutta automaattisen asennuksen vastaustiedoston pohjatiedostoa muokattiin (Ohjelmakoodi 13) siten, että rooliin asennettavaan laitteeseen ei asenneta työpöytäympäristöä.

Ohjelmakoodi 13. Automaattisen asennuksen vastaustiedoston Tasks-osio DEBIAN SERVER -laiteroolissa.

```
#
# Tasks
#
tasksel tasksel/first multiselect standard, laptop
d-i pkgsel/include string console-setup sudo
#
```

Tehtyjä muutoksia testattiin asentamalla laitteita eri rooleihin testiympäristön molemmista asennusverkoista. Asennukset onnistuivat moitteitta ja kaikki tehdyt muutokset toimivat suunnitellusti.

## 6 Johtopäätökset ja pohdinta

Muutoksilla saatiin aikaiseksi tilaajan vaatimukset täyttävä järjestelmä. Järjestelmästä pystytään asentamaan useita erilaisia laiterooleja ja laiteroolien ylläpitäminen on vaivatonta, eikä se vaadi ohjelmointityötä. Debian-käyttöjärjestelmän perushallinta ja automaattisen asennusprosessin tunteminen riittävät tarvittaviin ylläpitotoimiin. Tilaajan sovelluksille on erillinen pakettivarasto ja sen ylläpitämiseen on valmiit työkalut. Asennettavien laitteiden käyttäjätunnusten hallinta on piilotettu loppukäyttäjiltä. Laitteisiin määritettäviin käyttäjätunnuksiin ja salasanoihin voidaan vaikuttaa pohjatiedostoja muokkaamalla.

Muokattu järjestelmä on jo tilaajan tuotantokäytössä. Järjestelmään on lisätty erilaisia laiterooleja ja järjestelmää käyttäen on tehty yli 2000 Debian-laiteasennusta eri sijainneissa ja eri laiterooleihin. Ongelmia on ilmennyt hyvin vähän ja ne ovat pääosin liittyneet järjestelmän ulkopuolisiin (verkkoyhteydet yms.) toimintoihin. Järjestelmää ylläpitäviltä henkilöiltä ja loppukäyttäjiltä saatu palaute on ollut positiivista.

Tutkimustyön aikana Debian-käyttöjärjestelmästä julkaistiin uusi versio (11 / Bullseye). Uuden käyttöjärjestelmäversion asentamista järjestelmää käyttäen testattiin tutkimustyön päätteeksi. Uuden käyttöjärjestelmäversion tuominen järjestelmään oli vaivatonta ja vaati asennusmedioiden tuomisen lisäksi ainoastaan pieniä muutoksia automaattisen asennuksen vastaustiedoston pohjatiedostoihin.

Järjestelmää kehitetään edelleen uusien vaatimusten mukaiseksi. Jatkosuunnitelmissa on ainakin laitteiden levysalauksen toteuttaminen. Tutkimustyön tekijä on mukana tekemässä tätä jatkokehitystyötä.

## Lähteet

- Adelstein, T. & Lubanovic, B. (2007). *Linux System Administration*. Sebastopol, USA: O'Reilly Media, Inc.
- Debian. (n.d.). *Debian Wiki*. Haettu 18.10.2021 osoitteesta <https://wiki.debian.org/>
- Intel Corporation. (20.9.1999). *Preboot Execution Environment (PXE) Specification*. Haettu 3.3.2020 osoitteesta <http://www.pix.net/software/pxeboot/archive/pxespec.pdf>
- Kaario, K. (2002). *TCP/IP-verkot*. Jyväskylä: Docento Finland Oy.
- Microsoft. (2.10.2018). *Windows PE*. Haettu 25.2.2020 osoitteesta <https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-intro>
- Miradore. (7.1.2020). *Miradore Management Suite 5.2.0 Help* [käyttöohje]. Saatavilla vain Miradoren tukisivustolta. Vaatii tuotteen rekisteröinnin.
- Miradore. (n.d.). *Miradore Management Suite*. Haettu 25.2.2020 osoitteesta <https://mms.miradore.com/miradore-management-suite/>
- Network Working Group. (1992). *THE TFTP PROTOCOL (REVISION 2)*. Haettu 3.3.2020 osoitteesta <https://tools.ietf.org/pdf/rfc1350.pdf>
- Pollei, R. (2013). *Debian 7 : System Administration Best Practices*. Birmingham, United Kingdom: Packt Publishing.

**Liite 1: Alkuperäinen Debian.vbs -tiedosto**

```

#####
' Copyright 2011-2018 Miradore Ltd
'
' Create preseed file for Debian
'
' 20110914 PJN First Version
' 20111020 PJN Added apt-setup lines and netcfg/no_default_route
' 20111026 A H Set network fallback values even with DHCP
' 20120822 TTu Moved miradoreclient installation from apt-setup to postinstall
' 20130716 TTU Changed objFSO to g_objFSO
' 20180108 PJN UEFI support, changed C:\ to strTargetDir when creating latecmd.sh and postinst.sh
#####

ShowStatus "Creating installation profile ... ", False

Err.Clear

fUna.WriteLine "#"
fUna.WriteLine "# Repository"
fUna.WriteLine "#"
fUna.WriteLine "d-i mirror/country string manual"
fUna.WriteLine "d-i mirror/protocol select http"
fUna.WriteLine "d-i mirror/http/hostname string " & strHttpInstPointServer
fUna.WriteLine "d-i mirror/http/directory string " & strRepoPath
fUna.WriteLine "d-i mirror/http/proxy string"
fUna.WriteLine "d-i mirror/suite string " & OSEdition
fUna.WriteLine "d-i mirror/udeb/suite string " & OSEdition
fUna.WriteLine "d-i debian-installer/allow_unauthenticated boolean true"

Dim strRepoFolder
strRepoFolder = GetRepoFolder(SetupDrive & "_System\Miradore\Linux\Current", "Debian_" & OSVersion)
If strRepoFolder = "" Then
    ErrorHandler "There is no Miradore client repository for this version (" & OSVersion & ")." & vbCrLf & vbCrLf & _
        "OS installation will continue but the computer will not be managed by Miradore."
End If

' ## by default, d-i/apt-setup uses security and volatile as services here
' ## We can't include "volatile" here, because from 6.0 (squeeze), instead of volatile_host, it uses mirror
' ## configured above also with component "codename-updates" and DVD image doesn't include those updates
' ## "codename-updates" component from ftp.debian.org is included in post-install script
' ## security service still uses security_host , which can be used for those as default
fUna.WriteLine "d-i apt-setup/services-select select security"
fUna.WriteLine "d-i apt-setup/security_host string security.debian.org"

fUna.WriteLine "#"

```

```
fUna.WriteLine "#"  
fUna.WriteLine "# Partitioning"  
fUna.WriteLine "#"  
fUna.WriteLine "d-i partman-auto/disk string /dev/sda"  
fUna.WriteLine "d-i partman-lvm/device_remove_lvm boolean true"  
fUna.WriteLine "d-i partman-md/device_remove_md boolean true"  
fUna.WriteLine "d-i partman-auto/method string regular"  
fUna.WriteLine "d-i partman-auto/choose_recipe select atomic"  
fUna.WriteLine "d-i partman-partitioning/confirm_write_new_label boolean true"  
fUna.WriteLine "d-i partman/choose_partition select finish"  
fUna.WriteLine "d-i partman/confirm boolean true"  
fUna.WriteLine "d-i partman/confirm_nooverwrite boolean true"  
fUna.WriteLine "#"  
  
fUna.WriteLine "#"  
fUna.WriteLine "# Boot loader"  
fUna.WriteLine "#"  
fUna.WriteLine "d-i grub-installer/only_debian boolean true"  
fUna.WriteLine "d-i grub-installer/bootdev string default"  
  
fUna.WriteLine "#"  
fUna.WriteLine "# Regional settings"  
fUna.WriteLine "#"  
fUna.WriteLine "d-i debian-installer/locale string " & Replace(LanguageCode, "-", "_")  
fUna.WriteLine "d-i localechooser/supported-locales multiselect en_US.UTF-8" ' Make sure en_US is there allways  
  
If Len(Keymap) = 0 Then Keymap = "us"  
fUna.WriteLine "d-i debian-installer/keymap string " & Keymap  
fUna.WriteLine "d-i console-keymaps-at/keymap select " & Keymap  
fUna.WriteLine "d-i keyboard-configuration/xkb-keymap select " & LCase(Right(LanguageCode, 2))  
  
fUna.WriteLine "#"  
fUna.WriteLine "# Clock"  
fUna.WriteLine "#"  
fUna.WriteLine "d-i clock-setup/utc boolean true"  
fUna.WriteLine "d-i time/zone string " & TimeZoneName  
fUna.WriteLine "#"  
  
fUna.WriteLine "#"  
fUna.WriteLine "# Root user"  
fUna.WriteLine "#"  
fUna.WriteLine "d-i passwd/root-login boolean true"  
fUna.WriteLine "d-i passwd/root-password password " & AdminPassword  
fUna.WriteLine "d-i passwd/root-password-again password " & AdminPassword  
fUna.WriteLine "#"  
fUna.WriteLine "# Normal user"  
fUna.WriteLine "#"  
fUna.WriteLine "d-i passwd/make-user boolean false"  
'fUna.WriteLine "d-i passwd/user-fullname string Miradore user"  
'fUna.WriteLine "d-i passwd/username string miradore"
```

```

'fUna.WriteLine "d-i passwd/user-password password xxxxxxxx"
'fUna.WriteLine "d-i passwd/user-password-again password xxxxxxxx"

fUna.WriteLine "#"
fUna.WriteLine "# Network config"
fUna.WriteLine "#"
fUna.WriteLine "d-i netcfg/choose_interface select auto"
fUna.WriteLine "d-i netcfg/link_wait_timeout string 10"
If DHCP = "0" Then
    fUna.WriteLine "d-i netcfg/disable_dhcp boolean true"
    fUna.WriteLine "d-i netcfg/disable_autoconfig boolean true"
    fUna.WriteLine "d-i netcfg/dhcp_options select Configure network manually"
    fUna.WriteLine "d-i netcfg/get_nameservers string " & Replace(DNSServers, ",", " ")
    fUna.WriteLine "d-i netcfg/get_hostname string " & strCompName
    fUna.WriteLine "d-i netcfg/get_domain string " & DNSDomain
    fUna.WriteLine "d-i netcfg/get_ipaddress string " & IPAddress
    fUna.WriteLine "d-i netcfg/get_netmask string " & SubnetMask
    fUna.WriteLine "d-i netcfg/get_gateway string " & DefaultGateway
    fUna.WriteLine "d-i netcfg/confirm_static boolean true"
    fUna.WriteLine "d-i netcfg/no_default_route boolean true"
Else
    fUna.WriteLine "d-i netcfg/dhcp_timeout string 60"
    fUna.WriteLine "d-i netcfg/dhcpv6_timeout string 60"
End If
fUna.WriteLine "d-i netcfg/wireless_wep string"
fUna.WriteLine "#"

fUna.WriteLine "#"
fUna.WriteLine "# Tasks"
fUna.WriteLine "#"
Dim strTasksel
strTasksel = "tasksel tasksel/first multiselect standard, desktop"
If LCase(HWCategory) = "laptop" Then
    strTasksel = strTasksel & ", laptop"
End If
'strTasksel = strTasksel & ", gnome-desktop"
'strTasksel = strTasksel & ", kde-desktop"
'strTasksel = strTasksel & ", web-server"
'strTasksel = strTasksel & ", print-server"
'strTasksel = strTasksel & ", dns-server"
'strTasksel = strTasksel & ", file-server"
'strTasksel = strTasksel & ", mail-server"
'strTasksel = strTasksel & ", sql-database"
fUna.WriteLine strTasksel

' apt-transport-https needed to ensure https installation points work. But it is not on Debian disk 1
' fUna.WriteLine "d-i pkgselect/include string console-setup apt-transport-https"
fUna.WriteLine "d-i pkgselect/include string console-setup"

fUna.WriteLine "#"

```



```

fUna.WriteLine "# Pre-partman command"
fUna.WriteLine "#"
fUna.WriteLine "d-i partman/early_command string debconf-set partman-auto/disk ""$(list-devices disk | head -n1)""
fUna.WriteLine "#"

fUna.WriteLine "#"
fUna.WriteLine "# Post-commands"
fUna.WriteLine "#"

fUna.WriteLine "d-i preseed/late_command string chmod +x /latecmd.sh; /latecmd.sh"
fUna.WriteLine "#"

Dim fScriptFile
Err.Clear
Set fScriptFile = g_objFSO.CreateTextFile(strTargetDir & "\latecmd.sh", True)
If Err.Number = 0 Then

    WriteLinuxFile fScriptFile, "#!/bin/sh"
    WriteLinuxFile fScriptFile, "echo `date` === Late command script started === >> /var/log/latecmd.log"

    WriteLinuxFile fScriptFile, "echo `date` === Setting post-install script to be executed === >> /var/log/latecmd.log"
    WriteLinuxFile fScriptFile, "cp /postinst.sh /target/postinst.sh >> /var/log/latecmd.log"
    WriteLinuxFile fScriptFile, "chmod +x /target/postinst.sh >> /var/log/latecmd.log"
    WriteLinuxFile fScriptFile, "mv /target/etc/rc.local /target/etc/rc.local.orig"
    WriteLinuxFile fScriptFile, "echo '#!/bin/sh' > /target/etc/rc.local"
    WriteLinuxFile fScriptFile, "grep -F -v ""exit 0"" /target/etc/rc.local.orig >> /target/etc/rc.local"
    WriteLinuxFile fScriptFile, "echo /postinst.sh >> /target/etc/rc.local"
    WriteLinuxFile fScriptFile, "echo exit 0 >> /target/etc/rc.local"
    WriteLinuxFile fScriptFile, "chmod +x /target/etc/rc.local >> /var/log/latecmd.log"

    WriteLinuxFile fScriptFile, "echo `date` === Setting hostname === >> /var/log/latecmd.log"
    WriteLinuxFile fScriptFile, "oldhostname=`cat /target/etc/hostname`"
    WriteLinuxFile fScriptFile, "echo " & strCompName & ">/target/etc/hostname"
    WriteLinuxFile fScriptFile, "sed -i ""s/${oldhostname}/" & strCompName & "/g"" /target/etc/hosts >> /var/log/latecmd.log
2>&1"

    If LCase(Right(LanguageCode, 2)) <> "us" Then
        WriteLinuxFile fScriptFile, "echo `date` === Configuring keyboard === >> /var/log/latecmd.log"
        WriteLinuxFile fScriptFile, "sed -i ""s/XKBLayout=\"us\"/XKBLayout=\"\" & LCase(Right(LanguageCode, 2)) & \"\"/g""
/target/etc/default/console-setup >> /var/log/latecmd.log 2>&1"
        WriteLinuxFile fScriptFile, "sed -i ""s/XKBLayout=\"us\"/XKBLayout=\"\" & LCase(Right(LanguageCode, 2)) & \"\"/g""
/target/etc/default/keyboard >> /var/log/latecmd.log 2>&1"
    End If

    WriteLinuxFile fScriptFile, "echo `date` === Late command script ended === >> /var/log/latecmd.log"
    WriteLinuxFile fScriptFile, "exit 0"
    fScriptFile.Close
Else
    Writelog "ERROR: Failed to create " & strTargetDir & "\latecmd.sh (" & Err.Description & ")"
End If

```

```

Err.Clear
Set fScriptFile = g_objFSO.CreateTextFile(strTargetDir & "\postinst.sh", True)
If Err.Number = 0 Then

    WriteLinuxFile fScriptFile, "#!/bin/sh"
    WriteLinuxFile fScriptFile, "echo "`date` === Post-install script started ===""
    WriteLinuxFile fScriptFile, "echo "`date` === Post-install script started ===" >> /var/log/post-install.script.log"

    If DHCP = "0" Then

        WriteLinuxFile fScriptFile, "echo "`date` === Configuring resolv.conf ===" >> /var/log/post-install.script.log"
        WriteLinuxFile fScriptFile, "fgrep -v -e ""domain "" -e ""search "" /etc/resolv.conf >/etc/resolv.conf.new"
        WriteLinuxFile fScriptFile, "echo domain " & DNSDomain & ">>/etc/resolv.conf.new"
        If DNSSuffixList <> "" Then
            WriteLinuxFile fScriptFile, "echo search " & DNSDomain & " " & Replace(Replace(DNSSuffixList, ",", " "),
DNSDomain, "") & ">>/etc/resolv.conf.new"
        End If
        WriteLinuxFile fScriptFile, "cp /etc/resolv.conf /etc/resolv.conf.orig"
        WriteLinuxFile fScriptFile, "cp /etc/resolv.conf.new /etc/resolv.conf"

    Else

        If DNSSuffixList <> "" Then
            WriteLinuxFile fScriptFile, "if [ -d /etc/NetworkManager -a -x /etc/init.d/network-manager ];then"

            WriteLinuxFile fScriptFile, " echo "`date` === Stopping NetworkManager ===" >> /var/log/post-
install.script.log"
            WriteLinuxFile fScriptFile, " /etc/init.d/network-manager stop >> /var/log/post-install.script.log 2>&1"

            Dim strGUID : strGUID = CreateGuid()
            If strGUID = "" Then strGUID = "f6abecd0-3954-413d-8bb0-d57ef1787dc7"
            WriteLinuxFile fScriptFile, " echo "`date` === Configuring NetworkManager ===" >> /var/log/post-
install.script.log"
            WriteLinuxFile fScriptFile, " echo ""[connection]"" > /etc/NetworkManager/system-connections/Auto\ eth0"
            WriteLinuxFile fScriptFile, " echo ""id=Auto eth0"" >> /etc/NetworkManager/system-connections/Auto\ eth0"
            WriteLinuxFile fScriptFile, " echo ""uid=" & strGUID & """" >> /etc/NetworkManager/system-connections/Auto\
eth0"
            WriteLinuxFile fScriptFile, " echo ""type=802-3-ethernet"" >> /etc/NetworkManager/system-connections/Auto\
eth0"

            WriteLinuxFile fScriptFile, " echo ""autoconnect=true"" >> /etc/NetworkManager/system-connections/Auto\ eth0"
            WriteLinuxFile fScriptFile, " echo >> /etc/NetworkManager/system-connections/Auto\ eth0"
            WriteLinuxFile fScriptFile, " echo ""[ipv4]"" >> /etc/NetworkManager/system-connections/Auto\ eth0"
            WriteLinuxFile fScriptFile, " echo ""method=auto"" >> /etc/NetworkManager/system-connections/Auto\ eth0"
            WriteLinuxFile fScriptFile, " echo ""ignore-auto-dns=false"" >> /etc/NetworkManager/system-connections/Auto\
eth0"

            WriteLinuxFile fScriptFile, " echo ""dns-search=" & DNSDomain & " " & Replace(Replace(DNSSuffixList, ",", " "),
DNSDomain, "") & ";" >> /etc/NetworkManager/system-connections/Auto\ eth0"
            WriteLinuxFile fScriptFile, " chmod 600 /etc/NetworkManager/system-connections/Auto\ eth0"

```

```

WriteLinuxFile fScriptFile, " echo `date` === Starting NetworkManager === >> /var/log/post-install.script.log"
WriteLinuxFile fScriptFile, " /etc/init.d/network-manager start >> /var/log/post-install.script.log 2>&1"

WriteLinuxFile fScriptFile, " echo `date` === Waiting 10 seconds for NetworkManager to recreate
/etc/resolv.conf === >> /var/log/post-install.script.log"
WriteLinuxFile fScriptFile, " sleep 10"

WriteLinuxFile fScriptFile, "else"

WriteLinuxFile fScriptFile, " echo ""`date` === NetworkManager not in use, configuring resolv.conf directly
===>> /var/log/post-install.script.log"
WriteLinuxFile fScriptFile, " fgrep -v -e ""domain "" -e ""search "" /etc/resolv.conf >/etc/resolv.conf.new"
WriteLinuxFile fScriptFile, " echo domain " & DNSDomain & ">>/etc/resolv.conf.new"
If strDNSSearch <> "" Then
    WriteLinuxFile fScriptFile, " echo search " & strDNSSearch & ">>/etc/resolv.conf.new"
End If
WriteLinuxFile fScriptFile, " cp /etc/resolv.conf /etc/resolv.conf.orig"
WriteLinuxFile fScriptFile, " cp /etc/resolv.conf.new /etc/resolv.conf"

WriteLinuxFile fScriptFile, "fi"
End If

End If

WriteLinuxFile fScriptFile, "sleep 30"
WriteLinuxFile fScriptFile, "echo ""`date` === Creating Miradore client configuration ===>> /var/log/post-
install.script.log"
WriteLinuxFile fScriptFile, "echo # Miradore client configuration>/etc/miradore.conf"
Dim strClientParm
For Each strClientParm in Split(ClientParams, vbCrLf)
    WriteLinuxFile fScriptFile, "echo " & strClientParm & ">>/etc/miradore.conf"
Next

WriteLinuxFile fScriptFile, "echo ""`date` === Adding public repos ===>> /var/log/post-install.script.log"
WriteLinuxFile fScriptFile, "echo >>/etc/apt/sources.list"
WriteLinuxFile fScriptFile, "echo '# Added by Miradore initial install' >>/etc/apt/sources.list"
WriteLinuxFile fScriptFile, "echo deb http://ftp.debian.org/debian " & OSEdition & " main contrib>>/etc/apt/sources.list"
WriteLinuxFile fScriptFile, "echo deb http://ftp.debian.org/debian " & OSEdition & "-updates main
contrib>>/etc/apt/sources.list"
WriteLinuxFile fScriptFile, "echo >>/etc/apt/sources.list"

WriteLinuxFile fScriptFile, "echo ""`date` === Adding Miradore client repos ===>> /var/log/post-install.script.log"
WriteLinuxFile fScriptFile, "echo '# Miradore client repositories added by Miradore initial install'
>>/etc/apt/sources.list"
Dim strTmpRepo
i = 0
For Each strTmpRepo In arrHttpInstPoint
    WriteLinuxFile fScriptFile, "echo deb " & strTmpRepo & "/_System/Miradore/Linux/Current/" & strRepoFolder & " ./
>>/etc/apt/sources.list"
    WriteLog "Added repository " & strTmpRepo & "/_System/Miradore/Linux/Current/" & strRepoFolder & " ./"

```

Next

```
WriteLinuxFile fScriptFile, "apt-get update >> /var/log/post-install.script.log 2>&1"
```

```
WriteLinuxFile fScriptFile, "echo "`date` === Installing Miradore client ===" >> /var/log/post-install.script.log"
```

```
WriteLinuxFile fScriptFile, "apt-get install -y --force-yes miradoreclient >> /var/log/post-install.script.log 2>&1"
```

```
WriteLinuxFile fScriptFile, "echo "`date` === Cleaning script files ===" >> /var/log/post-install.script.log"
```

```
WriteLinuxFile fScriptFile, "rm /postinst.sh >> /var/log/post-install.script.log 2>&1"
```

```
WriteLinuxFile fScriptFile, "[ -e /etc/rc.local ] && rm /etc/rc.local"
```

```
WriteLinuxFile fScriptFile, "[ -e /etc/rc.local.orig ] && mv /etc/rc.local.orig /etc/rc.local"
```

```
WriteLinuxFile fScriptFile, "chmod -x /etc/rc.local"
```

```
WriteLinuxFile fScriptFile, "echo "`date` === Post-install script ended ===" >> /var/log/post-install.script.log"
```

```
WriteLinuxFile fScriptFile, "exit 0"
```

```
fScriptFile.Close
```

Else

```
Writelog "ERROR: Failed to create " & strTargetDir & "\postinst.sh (" & Err.Description & ")"
```

End If

```
fUna.WriteLine "#"
```

```
fUna.WriteLine "# Other"
```

```
fUna.WriteLine "#"
```

```
fUna.WriteLine "popularity-contest popularity-contest/participate boolean false"
```

```
fUna.WriteLine "d-i hw-detect/load_firmware boolean false"
```

```
fUna.WriteLine "d-i cdrom-detect/load_media boolean false"
```

```
fUna.WriteLine "d-i cdrom-detect/manual_config boolean false"
```

```
fUna.WriteLine "d-i finish-install/keep-consoles boolean true"
```

```
fUna.WriteLine "d-i finish-install/reboot_in_progress note"
```

```
fUna.WriteLine "#"
```

```
ShowStatus "Done", True
```

## Liite 2: Alkuperäinen preseed.cfg -tiedosto

```
#
# Repository
#
d-i mirror/country string manual
d-i mirror/protocol select http
d-i mirror/http/hostname string 10.100.10.11:80
d-i mirror/http/directory string /KUO/OS/Linux/Debian/10
d-i mirror/http/proxy string
d-i mirror/suite string buster
d-i mirror/udeb/suite string buster
d-i debian-installer/allow_unauthenticated boolean true
d-i apt-setup/services-select select security
d-i apt-setup/security_host string security.debian.org
#
#
# Partitioning
#
d-i partman-auto/disk string /dev/sda
d-i partman-lvm/device_remove_lvm boolean true
d-i partman-md/device_remove_md boolean true
d-i partman-auto/method string regular
d-i partman-auto/choose_recipe select atomic
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
#
#
# Boot loader
#
d-i grub-installer/only_debian boolean true
d-i grub-installer/bootdev string default
#
# Regional settings
#
d-i debian-installer/locale string fi_FI
d-i localechooser/supported-locales multiselect en_US.UTF-8
d-i debian-installer/keymap string fi-latin1
d-i console-keymaps-at/keymap select fi-latin1
d-i keyboard-configuration/xkb-keymap select fi
#
# Clock
#
d-i clock-setup/utc boolean true
d-i time/zone string Europe/Helsinki
#
#
```

```
# Root user
#
d-i passwd/root-login boolean true
d-i passwd/root-password password DefaultOSAdminPassword
d-i passwd/root-password-again password DefaultOSAdminPassword
#
# Normal user
#
d-i passwd/make-user boolean false
#
# Network config
#
d-i netcfg/choose_interface select auto
d-i netcfg/link_wait_timeout string 10
d-i netcfg/dhcp_timeout string 60
d-i netcfg/dhcpv6_timeout string 60
d-i netcfg/wireless_wep string
#
#
# Tasks
#
tasksel tasksel/first multiselect standard, desktop, laptop
d-i pkgsel/include string console-setup
#
# Pre-partman command
#
d-i partman/early_command string debconf-set partman-auto/disk "${list-devices disk | head -n1}"
#
#
# Post-commands
#
d-i preseed/late_command string chmod +x /latecmd.sh; /latecmd.sh
#
#
# Other
#
popularity-contest popularity-contest/participate boolean false
d-i hw-detect/load_firmware boolean false
d-i cdrom-detect/load_media boolean false
d-i cdrom-detect/manual_config boolean false
d-i finish-install/keep_consoles boolean true
d-i finish-install/reboot_in_progress note
#
```

**Liite 3: Alkuperäinen latecmd.sh -tiedosto**

```
#!/bin/sh
echo `date` === Late command script started === >> /var/log/latecmd.log
echo `date` === Setting post-install script to be executed === >> /var/log/latecmd.log
cp /postinst.sh /target/postinst.sh >> /var/log/latecmd.log
chmod +x /target/postinst.sh >> /var/log/latecmd.log
mv /target/etc/rc.local /target/etc/rc.local.orig
echo '#!/bin/sh' > /target/etc/rc.local
grep -F -v "exit 0" /target/etc/rc.local.orig >> /target/etc/rc.local
echo /postinst.sh >> /target/etc/rc.local
echo exit 0 >> /target/etc/rc.local
chmod +x /target/etc/rc.local >> /var/log/latecmd.log
echo `date` === Setting hostname === >> /var/log/latecmd.log
oldhostname=`cat /target/etc/hostname`
echo L0000005>/target/etc/hostname
sed -i "s/${oldhostname}/L0000005/g" /target/etc/hosts >> /var/log/latecmd.log 2>&1
echo `date` === Configuring keyboard === >> /var/log/latecmd.log
sed -i "s/XKBLayout=\"us\"/XKBLayout=\"fi\"/g" /target/etc/default/console-setup >>
/var/log/latecmd.log 2>&1
sed -i "s/XKBLayout=\"us\"/XKBLayout=\"fi\"/g" /target/etc/default/keyboard >>
/var/log/latecmd.log 2>&1
echo `date` === Late command script ended === >> /var/log/latecmd.log
exit 0
```

**Liite 4: Alkuperäinen postinst.sh -tiedosto**

```

#!/bin/sh
echo "`date` === Post-install script started ==="
echo "`date` === Post-install script started ===" >> /var/log/post-install.script.log
sleep 30
echo "`date` === Creating Miradore client configuration ===" >> /var/log/post-install.script.log
echo # Miradore client configuration>/etc/miradore.conf
echo InitialInstall=1>>/etc/miradore.conf
echo ServerInstanceGUID=0B97EDCA-7084-4E7D-931E-9BD15AE254DC>>/etc/miradore.conf
echo StartupDelay=60>>/etc/miradore.conf
echo PollingInterval=3600>>/etc/miradore.conf
echo FileLogSeverity=Info>>/etc/miradore.conf
echo SchedulerStartupDelay=120>>/etc/miradore.conf
echo ConfigUpdateInterval=1440>>/etc/miradore.conf
echo SchedulerPolling=3600>>/etc/miradore.conf
echo SchedulerTimeout=10800>>/etc/miradore.conf
echo ServerPermissionPolling=15>>/etc/miradore.conf
echo ServerInstanceName=AMUTEST>>/etc/miradore.conf
echo ServerHostName=10.100.10.10>>/etc/miradore.conf
echo ConnectionMethod=2>>/etc/miradore.conf
echo ClientListenPort=32227>>/etc/miradore.conf
echo IgnoreSSLErrors=1>>/etc/miradore.conf
echo ConfigurationID=1>>/etc/miradore.conf
echo Name=Miradore default server configuration>>/etc/miradore.conf
echo Priority=1>>/etc/miradore.conf
echo ProxyAuthMethod=none>>/etc/miradore.conf
echo ProxyHostName=Not set>>/etc/miradore.conf
echo ProxyPort=0>>/etc/miradore.conf
echo ProxyUserName=Not set>>/etc/miradore.conf
echo ServerHttpPort=80>>/etc/miradore.conf
echo SocketTimeout=30>>/etc/miradore.conf
echo UseProxy=0>>/etc/miradore.conf
echo ProxyPassword=Not set>>/etc/miradore.conf
echo >>/etc/miradore.conf
echo "`date` === Adding public repos ===" >> /var/log/post-install.script.log
echo >>/etc/apt/sources.list
echo '# Added by Miradore initial install' >>/etc/apt/sources.list
echo deb http://ftp.debian.org/debian buster main contrib>>/etc/apt/sources.list
echo deb http://ftp.debian.org/debian buster-updates main contrib>>/etc/apt/sources.list
echo >>/etc/apt/sources.list
echo "`date` === Adding Miradore client repos ===" >> /var/log/post-install.script.log
echo '# Miradore client repositories added by Miradore initial install' >>/etc/apt/sources.list
echo deb http://10.100.10.11:80/KUO/_System/Miradore/Linux/Current/Debian_10 ./
>>/etc/apt/sources.list
apt-get update >> /var/log/post-install.script.log 2>&1
echo "`date` === Installing Miradore client ===" >> /var/log/post-install.script.log
apt-get install -y --force-yes miradoreclient >> /var/log/post-install.script.log 2>&1
echo "`date` === Cleaning script files ===" >> /var/log/post-install.script.log

```



```
rm /postinst.sh >> /var/log/post-install.script.log 2>&1
[ -e /etc/rc.local ] && rm /etc/rc.local
[ -e /etc/rc.local.orig ] && mv /etc/rc.local.orig /etc/rc.local
chmod -x /etc/rc.local
echo "`date` === Post-install script ended ===" >> /var/log/post-install.script.log
exit 0
```

## Liite 5: Pakettivaraston allekirjoitusavaimien luominen

OpenPGP-avaimien luominen. Asetusten hyväksymisen jälkeen kysytään salasana salaiselle avaimelle.

```
10.100.10.11 - amu@INSTP-KUO: ~ VT
File Edit Setup Control Window Help
root@INSTP-KUO:~# gpg --full-generate-key
gpg (GnuPG) 2.2.12; Copyright (C) 2018 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Valitse millaisen avaimen haluat:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (vain allekirjoitus)
  (4) RSA (vain allekirjoitus)
Valintasi? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Halutun avaimen koko on 4096 bittiä
Kuinka kauan avaimen tulee olla voimassa.
  0 = Avain ei vanhene koskaan
  <n> = Avain vanhenee n päivän kuluttua
  <n>w = Avain vanhenee n viikon kuluttua
  <n>m = Avain vanhenee n kuukauden kuluttua
  <n>y = Avain vanhenee n vuoden kuluttua
Avain on voimassa? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Oikea nimi: Ari Mustonen
Sähköpostiosoite: ari@firmaxyz.x
Huomautus: Repositoryn allekirjoitusavain
Valitsit seuraavan käyttäjätunnuksen:
  "Ari Mustonen (Repositoryn allekirjoitusavain) <ari@firmaxyz.x>"
Muuta (N)imi, (H)uomautus, (S)ähköposti vai (O)k/(L)opeta?0
```

Avaimelle luotiin lisäksi erillinen allekirjoitusavain.

```
10.100.10.11 - amu@INSTP-KUO: ~ VT
File Edit Setup Control Window Help

root@INSTP-KUO:/home/amu# gpg --edit-key ari@firmaxyz.x
gpg (GnuPG) 2.2.12; Copyright (C) 2018 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Salainen avain on saatavilla.

sec  rsa4096/CDCFBCC93872BC8C
    created: 2021-10-15  expires: ei koskaan  usage: SC
    trust: ultimate      validity: ultimate
ssb  rsa4096/F7769547D757D881
    created: 2021-10-15  expires: ei koskaan  usage: E
[ultimate] (1). Ari Mustonen (Repositoryn allekirjoitusavain) <ari@firmaxyz.x>

gpg> addkey
Valitse millaisen avaimen haluat:
  (3) DSA (vain allekirjoitus)
  (4) RSA (vain allekirjoitus)
  (5) Elgamal (encrypt only)
  (6) RSA (vain salaus)
Valintasi? 4
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Halutun avaimen koko on 4096 bittiä
Kuinka kauan avaimen tulee olla voimassa.
  0 = Avain ei vanhene koskaan
  <n> = Avain vanhenee n päivän kuluttua
  <n>w = Avain vanhenee n viikon kuluttua
  <n>m = Avain vanhenee n kuukauden kuluttua
  <n>y = Avain vanhenee n vuoden kuluttua
Avain on voimassa? (0) 0
Key does not expire at all
Is this correct? (y/N) y
Really create? (y/N) y
Tarvitaan paljon satunnaislukuja. Voit suorittaa muita toimintoja
(kirjoittaa näppäimistöllä, liikuttaa hiirtä, käyttää levyjä)
alkulukujen luomisen aikana, tämä antaa satunnaislukugeneraattorille
paremmat mahdollisuudet kerätä riittävästi entropiaa.

sec  rsa4096/CDCFBCC93872BC8C
    created: 2021-10-15  expires: ei koskaan  usage: SC
    trust: ultimate      validity: ultimate
ssb  rsa4096/F7769547D757D881
    created: 2021-10-15  expires: ei koskaan  usage: E
ssb  rsa4096/3BB6B203CAC42931
    created: 2021-10-15  expires: ei koskaan  usage: S
[ultimate] (1). Ari Mustonen (Repositoryn allekirjoitusavain) <ari@firmaxyz.x>

gpg> save
root@INSTP-KUO:/home/amu#
```

Tarvittavat avaimet luotu. Merkki "S" avaimen perässä tarkoittaa allekirjoitusavainta.

```
10.100.10.11 - amu@INSTP-KUO: ~ VT
File Edit Setup Control Window Help
root@INSTP-KUO:/home/amu/keys# gpg -k
/root/.gnupg/pubring.kbx
-----
pub   rsa4096 2021-10-15 [SC]
      86EA73DDDAC7A5235C2B2DD7CDCFBCC93872BC8C
uid           [ultimate] Ari Mustonen (Repositoryn allekirjoitusavain) <ari@firmaxyz.x>
sub   rsa4096 2021-10-15 [E]
sub   rsa4096 2021-10-15 [S]

root@INSTP-KUO:/home/amu/keys#
```

Lopuksi vietiin julkinen avain ASCII-tiedostoon (firmaxyz.public.key). Samassa yhteydessä otettiin myös varmuuskopio salaisesta avaimesta. Varmuuskopion avulla voidaan pakettivaraston allekirjoittaminen tehdä tarvittaessa toisella laitteella.

```
10.100.10.11 - amu@INSTP-KUO: ~ VT
File Edit Setup Control Window Help
root@INSTP-KUO:/home/amu/keys# gpg --output ./firmaxyz.public.key --armor --export ari@firmaxyz.x
root@INSTP-KUO:/home/amu/keys# gpg --output ./firmaxyz.secret.gpg --export-secret-keys ari@firmaxyz.x
root@INSTP-KUO:/home/amu/keys# ls -la
yhteensä 24
drwx----- 2 root root 4096 loka 15 13:19 .
drwx----- 4 amu amu 4096 loka 15 11:41 ..
-rw----- 1 root root 5422 loka 15 13:17 firmaxyz.public.key
-rw----- 1 root root 7956 loka 15 13:19 firmaxyz.secret.gpg
root@INSTP-KUO:/home/amu/keys#
```

**Liite 6: Shell-skripti pakettivaraston julkaisemiseen**

```
# /usr/local/bin/mir-release-repository
#!/bin/bash

# Ari Mustonen, INTIM19A6
# Suorita repositoryn juuressa (/var/miradore/setup/Repository)
# Luo Packages ja Release fileit

set -e

# Suoritus repositoryn juuressa (siellä, missä pool-hakemisto)
if [ ! -d "pool" ] ; then
    echo "Olet väärässä hakemistossa!"
    exit 1
fi

# Component luetaan scriptille annetusta parametrasta
# Ilman parametria on 'main'
if [ "$#" -ne "1" ]
then
    COMPONENT="main"
else
    COMPONENT=$1
fi

# DIST vrt. buster, mutta saa olla ihan oma "jakeluversio" nimi
DIST="xyz"

# Vapaamuotoinen teksti, alkuperäinen lähde paketeille
ORIGIN="Firma XYZ"

# Kuvaava teksti
LABEL="Omat sovellukset"

# Arkkitehtuurit
ARCHS="amd64"

mkdir -p dists/${DIST}

for ARCH in ${ARCHS} ; do
    echo "Skannataan paketit komponentille ${COMPONENT} ja arkkitehtuurille ${ARCH}"
    mkdir -p dists/${DIST}/${COMPONENT}/binary-${ARCH}
    dpkg-scanpackages -m -a ${ARCH} pool/${COMPONENT} > dists/${DIST}/${COMPONENT}/binary-
${ARCH}/Packages
    gzip -c dists/${DIST}/${COMPONENT}/binary-${ARCH}/Packages >
dists/${DIST}/${COMPONENT}/binary-${ARCH}/Packages.gz
done
```

```
echo "Luodaan Release-tiedosto"
```

```
REL_COMPONENTS=`find pool -maxdepth 1 -mindepth 1 -type d | xargs -L1 basename | xargs echo -n`  
RELEASECONF=/tmp/release.conf
```

```
cat << EOF > ${RELEASECONF}  
APT::FTPArchive::Release::Codename "${DIST}";  
APT::FTPArchive::Release::Origin "${ORIGIN}";  
APT::FTPArchive::Release::Components "${REL_COMPONENTS}";  
APT::FTPArchive::Release::Label "${LABEL}";  
APT::FTPArchive::Release::Architectures "${ARCHS}";  
APT::FTPArchive::Release::Suite "stable";  
EOF
```

```
cd dists/${DIST}  
apt-ftparchive -c ${RELEASECONF} release . > Release  
cd ../../
```

```
chown -R mir-writer:instp dists
```

```
echo "Release-tiedosto valmis"  
echo "MUISTA ALLEKIRJOITTA REPOSITORY!"  
echo
```

**Liite 7: Shell-skripti pakettivaraston julkaisutiedostojen allekirjoittamiseen**

```
# /usr/local/bin/mir-sign-release
#!/bin/bash

# Ari Mustonen, INTIM19A6
# Repositoryn allekirjoitus
# Allekirjoitusavaimen id pitää antaa parametrina

# Avaimet nakee komennolla:
# gpg -k

# Suoritus repositoryn juuressa (siellä, missä dists-hakemisto)
if [ ! -d "dists" ] ; then
    echo "Olet väärässä hakemistossa!"
    exit 1
fi

if [ "$#" -ne "1" ]; then
    echo "Käyttö: $0 <keyid>"
    exit 1
fi

KEYID=$1

read -s -p "Avaimen salasana (ei näy kirjoitettaessa): " PW
echo

export GPG_TTY=$(tty)

if [ -n "${PW}" ] ; then
    # Testataan salasanaa...
    echo "onnistuukohan" | gpg --batch -o /dev/null -u "${KEYID}" --passphrase ${PW} --pinentry-
mode loopback --detach-sign - && echo "HyvinMeni" | grep "HyvinMeni" > /dev/null
    RES=$?
    if [ "$RES" -ne "0" ]; then
        echo
        echo "Väärä salasana!"
        echo
        exit 1
    fi
    echo
    echo "Salasana OK. Jatketaan..."
fi

echo
echo "Luodaan Release.gpg..."
```

```
find . -type f -name "Release" -printf '%h\n' | xargs -I{} gpg --batch --digest-algo SHA256 --yes
-v -u "${KEYID}" --passphrase ${PW} --pinentry-mode loopback --output {}/Release.gpg --detach-
sign {}/Release
echo
echo "Luodaan InRelease..."
find . -type f -name "Release" -printf '%h\n' | xargs -I{} gpg --batch --digest-algo SHA256 --yes
-v -u "${KEYID}" --passphrase ${PW} --pinentry-mode loopback --output {}/InRelease --clear-sign
{}/Release

echo
echo RELOADAGENT | gpg-connect-agent
echo
```



**Liite 8: Shell-skripti salasana-tiedoston luomiseen**

```
# /usr/local/bin/mir-create-pwdfile
# Ari Mustonen, INTIM19A6

#!/bin/bash

# Kysyy perus- ja ylläpitokäyttäjien salasanan
# Kirjoittaa salasanat passwords.pwd -tiedostoon
# Vaatii paketin whois (sisältää mkpasswd) asennettuna

clear
echo
echo "Skripti kysyy salasanat ja kirjoittaa ne salattuina passwords.pwd -tiedostoon"
echo "Vaihda tiedoston nimi vastaamaan laiteroolia"

echo
echo "peruskäyttäjän salasana:"
read user_pwd
echo
echo "ylläpitokäyttäjän salasana:"
read admin_pwd

mkpasswd --method=SHA-512 $user_pwd > passwords.pwd
mkpasswd --method=SHA-512 $admin_pwd >> passwords.pwd

echo >> passwords.pwd
echo "Tiedosto luotu 'mir-create-pwdfile' skriptillä." >> passwords.pwd
echo "ÄLÄ MUOKKAA TIEDOSTOA MUILLA TAVOILLA!" >> passwords.pwd
echo
```

**Liite 9: Lopullinen Debian.vbs -tiedosto**

```
' /var/miradore/setup/_System/_Start/Debian.vbs
' Ari Mustonen, INTIM19A6

ShowProgress "WriteLine", ""

Dim TemplateName, PreseedTemplate, LateTemplate, PostTemplate, strContent
Dim PwdFileName, PwdFile, uPwd, aPwd

' Funktio jolla muokataan pohjatiedostoja
Sub ReplaceVariable(strContent, template, value)
    strContent = Replace(strContent, template, value, 1, -1, 1)
End Sub

' Korvataan laiteroolin mahdollinen välilyönti väliviivalla
TemplateName = Replace(DeviceRole, " ", "-")

' Määritetään pohjatiedostojen sijainti laitteen roolin (DEVICE ROLE) mukaan
PreseedTemplate = SetupDrive & "\Templates\" & TemplateName & "\preseed.cfg"
LateTemplate = SetupDrive & "\Templates\" & TemplateName & "\latecmd.sh"
PostTemplate = SetupDrive & "\Templates\" & TemplateName & "\postinst.sh"

' Luetaan preseed.cfg templaatti muistiin
ShowProgress "Write", "- Reading preseed.cfg template... "
Set fPreseed = g_objFSO.OpenTextFile(PreseedTemplate)
strContent = fPreseed.ReadAll
fPreseed.Close
ShowProgress "WriteLine", "Done"

' Muokataan preseed -pohjatiedoston muuttujat asennusvalintojen mukaisiksi
ShowProgress "Write", "- Replacing template variables... "
ReplaceVariable strContent, "_HTTPINSTPOINTSERVER_", strHttpInstPointServer
ReplaceVariable strContent, "_INSTALLERPWD_", AdminPassword
ReplaceVariable strContent, "_COMPNAME_", strCompName
ShowProgress "WriteLine", "Done"

' Kirjoitetaan preseed MDInst.vbs käyttämään muuttujaan (fUna)
fUna.Write(strContent)

' Luetaan latecmd.sh -pohjatiedosto muistiin
ShowProgress "Write", "- Reading latecmd.sh template... "
Set fLateTemplate = g_objFSO.OpenTextFile(LateTemplate)
strContent = fLateTemplate.ReadAll
fLateTemplate.Close
ShowProgress "WriteLine", "Done"

' Luetaan salasanat lokalisointitiedostosta
PwdFileName = TemplateName & ".pwd"
```

```

If g_objFSO.FileExists(SetupDrive & "\\Localization\" & LocSN & "\" & PwdFileName) Then
    PwdFile = SetupDrive & "\\Localization\" & LocSN & "\" & PwdFileName
    ShowProgress "WriteLine", "- Localization file found for location (" & LocSN & ")."
ElseIf g_objFSO.FileExists(SetupDrive & "\\Localization\DEFAULT\" & PwdFileName) Then
    PwdFile = SetupDrive & "\\Localization\DEFAULT\" & PwdFileName
    ShowProgress "WriteLine", "- Localization file not found. Using DEFAULT."
Else
    ShowProgress "WriteLine", "- ERROR! Localization file not found!"
End If

Set fPwd = g_objFSO.OpenTextFile(PwdFile)
uPwd = fPwd.ReadLine
aPwd = fPwd.ReadLine
fPwd.Close

' Kirjoitetaan (korvataan) salasanat muistissa olevaan pohjatiedostoon
ReplaceVariable strContent, "_USERPWD_", uPwd
ReplaceVariable strContent, "_ADMINPWD_", aPwd

' Luodaan uusi latecmd.sh laitteelle
ShowProgress "Write", "- Editing latecmd.sh... "
Set fLateFile = g_objFSO.CreateTextFile(strTargetDir & "\\latecmd.sh", True)

' Kirjoitetaan pohjatiedoston sisältö tiedostoon
fLateFile.Write(strContent)

' Luodaan Miradore clientin konffifile
WriteLinuxFile fLateFile, "echo # Miradore client configuration > /etc/miradore.conf"
Dim strClientParm
For Each strClientParm in Split(ClientParams, vbCrLf)
    WriteLinuxFile fLateFile, "echo " & strClientParm & " >> /etc/miradore.conf"
Next

' Miradore clientin asennus
WriteLinuxFile fLateFile, "apt-get install --yes miradoreclient >> /var/log/latecmd.log"

' Postinst.sh suorittaminen
WriteLinuxFile fLateFile, "bash /tmp/postinst.sh"
WriteLinuxFile fLateFile, "exit 0"

' Suljetaan latecmd.sh
fLateFile.Close
ShowProgress "WriteLine", "Done"

' Kopioidaan postinstall.sh asennuspisteeltä asennettavalle laitteelle
ShowProgress "Write", "- Reading postinstall.sh template... "
g_objFSO.CopyFile PostTemplate, strTargetDir & "\\postinst.sh"
ShowProgress "WriteLine", "Done"

```

**Liite 10: Lopullinen preseed.cfg -pohjatiedosto**

```
# preseed.cfg -pohjatiedosto
# /var/miradore/setup/Teplates/DEBIAN/preseed.cfg
# Ari Mustonen, INTIM19A6
#
#   Repository
#
d-i mirror/country string manual
d-i mirror/protocol select http
d-i mirror/http/hostname string _HTTPINSTPOINTSERVER_
d-i mirror/http/directory string /Debian/10
d-i mirror/http/proxy string
d-i mirror/suite string buster
d-i mirror/udeb/suite string buster
d-i apt-setup/local0/repository string deb http://_HTTPINSTPOINTSERVER_/Repository xyz main
d-i apt-setup/local0/source boolean false
d-i apt-setup/local0/key string http://_HTTPINSTPOINTSERVER_/Repository/firmaxyz.public.key
d-i debian-installer/allow_unauthenticated boolean false
d-i apt-setup/services-select select none
#
#
#   Partitioning
#
d-i partman-auto/disk string /dev/sda
d-i partman-lvm/device_remove_lvm boolean true
d-i partman-md/device_remove_md boolean true
d-i partman-auto/method string regular
d-i partman-auto/choose_recipe select atomic
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
#
#
#   Boot loader
#
d-i grub-installer/only_debian boolean true
d-i grub-installer/bootdev string default
#
#   Regional settings
#
d-i debian-installer/locale string fi_FI
d-i localechooser/supported-locales multiselect en_US.UTF-8
d-i debian-installer/keymap string fi-latin1
d-i console-keymaps-at/keymap select fi-latin1
d-i keyboard-configuration/xkb-keymap select fi
#
```

```
# Clock
#
d-i clock-setup/utc boolean true
d-i time/zone string Europe/Helsinki
d-i clock-setup/ntp boolean false
#
#
# Root user
#
d-i passwd/root-login boolean false
#
# Normal user
#
d-i passwd/make-user boolean false
d-i passwd/user-fullname string installer
d-i passwd/username string installer
d-i passwd/user-password password _INSTALLERPWD_
d-i passwd/user-password-again password _INSTALLERPWD_
#
# Network config
#
d-i netcfg/choose_interface select auto
d-i netcfg/get_nameservers string
d-i netcfg/get_hostname string _COMPNAME_
d-i netcfg/link_wait_timeout string 10
d-i netcfg/dhcp_timeout string 60
d-i netcfg/dhcpv6_timeout string 60
d-i netcfg/wireless_wep string
#
#
# Tasks
#
tasksel tasksel/first multiselect standard, laptop
d-i pkgsel/include string console-setup sudo task-gnome-desktop
#
# Pre-partman command
#
d-i partman/early_command string debconf-set partman-auto/disk "${list-devices disk | head -n1}"
#
#
# Post-commands
#
d-i preseed/late_command string cp /postinst.sh /target/tmp/postinst.sh; cp /latecmd.sh
/target/tmp/latecmd.sh; chroot /target sh -c "/bin/bash /tmp/latecmd.sh"
#
#
# Other
#
popularity-contest popularity-contest/participate boolean false
d-i hw-detect/load_firmware boolean false
```

```
d-i cdrom-detect/load_media boolean false
d-i cdrom-detect/manual_config boolean false
d-i finish-install/keep-consoles boolean true
d-i finish-install/reboot_in_progress note
#
```

**Liite 11: Lopullinen latecmd.sh –pohjatiedosto**

```
# latecmd.sh -pohjatiedosto
# /var/miradore/setup/Templates/DEBIAN/latecmd.sh
# Ari Mustonen, INTIM19A6

#!/bin/sh
echo "`date` === latecmd script started ===" >> /var/log/latecmd.log

# Luodaan käyttäjätilit ja määritetään salasanat
useradd -m -p '_USERPWD_' -s /bin/bash user
usermod -aG cdrom,audio,video,netdev,bluetooth user
useradd -m -p '_ADMINPWD_' -s /bin/bash admin
usermod -aG sudo,netdev admin

# Skriptiä täydennetään Debian.vbs:n suorituksen aikana
```

**Liite 12: Lopullinen postinst.sh –pohjatiedosto**

```
# postinst.sh -pohjatiedosto
# /var/miradore/setup/Templates/DEBIAN/postinst.sh
# Ari Mustonen, INTIM19A6

#!/bin/sh
echo "`date` === postinst script started ===" >> /var/log/postinst.log

# Tähän alle kolmannen osapuolen kovenukset yms.
# Tapahtumalokien kirjoitus tiedostoon /var/log/postinst.log

exit 0
```