Perttu Pisilä

ANDROID-PUHELIMEN FIRMWARE-ETÄPÄIVITYS

ANDROID-PUHELIMEN FIRMWARE-ETÄPÄIVITYS

Perttu Pisilä
Opinnäytetyö
Kevät 2014
Tietotekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, langattomat laitteet

Tekijä: Perttu Pisilä
Opinnäytetyön nimi: Android-puhelimen firmware-etäpäivitys
Työn ohjaaja: Lauri Pirttiaho
Työn valmistumislukukausi ja -vuosi: Kevät 2014
Sivumäärä: 35 + 2 liitettä

Opinnäytetyön aiheena oli kartoittaa vaihtoehtoisia menetelmiä Anite Finland Oy:n asiakkaiden Android-puhelimien kentällä tapahtuvan firmware-päivitysprosessin parantamiseksi. Kartoitetuista menetelmistä tuli valita paras ja esittää se tilaajalle. Tavoitteena oli parantaa nykyistä menettelyä, joka sisältää useita ongelmakohtia. Uuden menetelmän oli tuettava Samsung-merkkisiä Android-laitteita.

Työ suoritettiin etsimällä tietoa internet-lähteistä sekä keskustelemalla vaihtoehtoisista menetelmistä tilaajayrityksen asiantuntijoiden kanssa. Päätös valitusta menetelmästä tehtiin näiden keskusteluiden pohjalta. Työ laajennettiin koskemaan myös vaatimusmäärittelyn luomista, jonka avulla menetelmän kehittämisestä pyydettiin tarjouspyyntö alihankkijalta.

Valittu menetelmä esitettiin tilaajalle, joka päättää sen käyttöönottamisesta saatuaan alihankkijalta toteutuksen kustannusarvion. Uusi menetelmä sisältää lukuisia parannuksia firmware-etäpäivitysprosessiin ja on tulevaisuudessa laajennettavissa myös muita toimintoja varten.

Asiasanat: Android, firmware, päivitys

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, option of Wireless Devices

---

Author: Perttu Pisilä
Title of thesis: Android firmware remote update system
Supervisor: Lauri Pirttiaho
Term and year when the thesis was submitted: Spring 2014
Pages: 35 + 2 appendices

The objective of this thesis is to research and evaluate alternative methods and find the best solution for updating customer Android-device firmware on field. Anite Finlad offers these firmware updates as a part of their Nemo Handy-A wireless network measurement software. The new remote update system has to support Samsung's Android devices.

The research is executed using online information sources and discussions with the experts in the assigner company. Evaluation and the final decision are based on these discussions. A specification document is also produced.

The chosen method is demonstrated to assigner, which makes the decision of starting the development based on the tender from the subcontractor. The new remote update method offers multiple improvements compared to the current method. There are also multiple possible features that might be added in the future.

Keywords: Android, Firmware, Update

ALKULAUSE

Anite Finland tarjosi erittäin mielenkiintoisen mahdollisuuden päästä tutkimaan ja kehittämään ratkaisua käytännön ongelmaan firmware-etäpäivitysprosessissa.

Haluaisin kiittää työn ohjaajaa, Sauli Mönttistä, sekä kaikkia muita mukana olleita Nemolaisia. Oli hienoa päästä työskentelemään ammattitaitoisten ihmisten parissa.

Kiitos Tuula Hopeavuorelle tekstinohjauksesta. Kiitos myös ohjaavalle opettajalle, Lauri Pirttiaholle. Säännöllinen palaute auttoi työn etenemisessä ja aikataulussa pysymisessä.

Oulussa 10.4.2014

Perttu Pisilä

SISÄLLYS

1 JOHDANTO

Anite Finland Oy, entinen Nemo Technologies Oy, on osa Anite-konsernia. Yritys kehittää langattomien tietoliikenneverkkojen mittausjärjestelmiä ja analysointiohjelmistoja (1). Yrityksen Nemo-tuoteperheeseen kuuluvia tuotteita ovat mm. Outdoor, Analyze ja Handy.

Nemo Outdoor on langattomien verkkojen mittauksiin käytettävä työkalu. Tyypillisessä Outdoor-mittausjärjestelmässä ohjelmisto toimii kannettavassa tietokoneessa, johon on kytketty yksi tai useampi verkkolaite, kuten matkapuhelin tai RF-skanneri. Nemo Handy on puolestaan älypuhelimessa ajettava mittausohjelmisto. Nemo Analyze on mittaustulosten analysointiohjelmisto, jolla mittaustuloksia voidaan tarkastella ja jolla niistä voidaan luoda raportteja. Lisää tietoa Nemo-tuoteperheestä löytyy Aniten internet-sivuilta (2). Nemo Outdoor ja Analyze käyttävät USB-avaimeen pohjautuvaa kopionsuojausmenetelmää.

Opinnäytetyön aiheeksi etsittiin yrityksen sisältä asiaa, johon kaivattiin parannusta ja joka olisi aiheeltaan opinnäytetyön suorittajalle mieleinen sekä laajuudeltaan ja aikataulultaan opinnäytetyön vaatimuksiin sopiva. Useiden eri henkilöiden mielestä toimituksen jälkeiseen kentällä tapahtuvaan asiakaslaitteen (puhelimen) firmware-etäpäivitykseen kaivattiin parempaa menetelmää. Erityisesti kyseessä on Handy-A-mittausohjelmistoa käyttävien Android-laitteiden ongelma.

Android on avoimeen lähdekoodin (open source) Linux-ytimeen perustuva mobiilikäyttöjärjestelmä. Androidin alkuperäinen kehittäjä on Android Inc -yritys, jonka Google osti vuonna 2005 (3). Android on kasvattanut suosiotaan nopeasti älypuhelimissa. Vuoden 2013 toisella neljänneksellä Android oli käyttöjärjestelmänä lähes 80%:ssa myydyistä älypuhelimista (4).

Suurin Android-älypuhelimien valmistaja on eteläkorealainen Samsung. Samsungin osuus myydyistä Android-mobiililaitteista vuoden 2013 marraskuussa oli 63% (5). Samsung Electronics on maailman suurin elektroniikka-alan yritys (6) ja suurin matkapuhelinten valmistaja ohitettuaan Nokian toimitusmäärissä vuoden 2012 ensimmäisellä neljänneksellä (7). Samsungilla on vahva osuus myös

Aniten toimittamista puhelimista, joten tämä opinnäytetyö rajattiin koskemaan vain Samsung-merkkisten Android-puhelinten etäpäivityksiä.

Nemo Handy-A on Android-älypuhelimeen asennettava ohjelmisto, jonka kautta voidaan suorittaa puheluita, tekstiviestejä, tiedonsiirtoja ja muita toimintoja manuaalisesti tai skriptin ohjaamana. Ohjelmisto kerää lokia puhelimen verkkorajapinnan tapahtumista myöhempää analysointia varten (8). Handy-ohjelmiston toiminnan kannalta puhelimeen vaaditaan pääkäyttäjän (root, superuser) oikeudet.

Toimenpidettä, jossa Android-järjestelmään hankitaan pääkäyttäjän oikeudet, kutsutaan roottaamiseksi (rooting). Pääkäyttäjän oikeuksilla käyttäjä tai puhelimeen asennettu ohjelma pystyy lukemaan ja muokkaamaan kaikkia puhelimen tiedostoja. Puhelimeen on silloin myös mahdollista asentaa erikoisohjelmistoja, jotka tarvitsevat pääsyn syvemmälle järjestelmän tietoihin. Tällaisia ohjelmia voivat olla esimerkiksi tietoturva- tai mittausohjelmistot. (9.)

Elektroniikkalaitteen kuten älypuhelimen laiteohjelmisto eli firmware on tallennettu laitteen haihtumattomaan muistiin. Firmwaresta käytetäänkin usein myös hieman harhaanjohtavaa termiä ROM, joka viittaa sen sijaitsevan lukumuistissa (Read Only Memory). Lukumuistissa olevaa ohjelmistoa ei yleensä muokata ajon aikana, vaan sen päivittäminen vaatii erityistoimenpiteitä. Useissa nykyaikaisissa laitteissa firmware sijaitsee flash-tyyppisessä muistissa, jota on mahdollisuus kirjoittaa uudelleen ohjelmallisesti. Tästä syntyykin yleisesti firmwarepäivityksestä käytetty termi flashing. Niin sanotun flashayksen yhteydessä voidaan päivittää koko Android-järjestelmä tai vain joitain sen osia, kuten pelkkä käynnistyslatain. (10.)

Ensimmäinen Android-järjestelmässä käynnistyvä osa on käynnistyslatain eli bootloader. Se on laitteistokohtainen ohjelma, jonka tehtävänä on mm. käynnistää käyttöjärjestelmän ydin eli kernel. Bootloader on uusissa puhelimissa usein lukittu, jolloin se lataa puhelimeen ainoastaan laitteen valmistajan hyväksymän firmware-ohjelmiston. Lukitus voidaan kuitenkin purkaa, jolloin laitteeseen voidaan hankkia pääkäyttäjän oikeudet ja asentaa kolmannen osapuolen firmware. (11.)

Bootloaderin jälkeen käynnistyy käyttöjärjestelmän ydin eli kernel. Kernel toimii laitteen laitteiston ja ohjelmiston välissä ohjaten esimerkiksi kosketusnäytölle syötettävät komennot käynnissä olevalle ohjelmalle ja ohjelman syöttämät tiedot näytölle. Kernelin päälle käynnistyy varsinainen käyttöjärjestelmä ja suoritettavat ohjelmat. (12.)

Anite asentaa toimittamiinsa puhelimiin oman firmware-ohjelmiston, jonka tehtävänä on luoda mittausohjelmistoille pääsy laitteen verkkorajapinnan tapahtumiin. Samsung-laitteissa tämä päivitys tehdään Samsungin Odin3-työkalulla, joka käyttää päivityksiin tar-muotoon pakattuja järjestelmäkuvia (system image). Päivitys voi sisältää koko järjestelmän tai esimerkiksi pelkän ytimen. Kokonaisen järjestelmäkuvan koko on noin 2 GB.

Toistaiseksi myös etäpäivitykset on suoritettu samalla Odin-työkalulla, mutta yhteys asiakkaaseen on otettu USB-tunnelin välityksellä. Menetelmässä Odin toimii paikallisessa tietokoneessa ja sen data kulkee internetin kautta virtuaalisen USB-tunnelin välityksellä asiakkaan tietokoneeseen ja puhelimeen. Syynä menettelyyn on ollut se, ettei muokattuja firmware-tiedostoja ole haluttu luovuttaa asiakkaille.

Kyseinen menetelmä on kuitenkin hidas ja epäluotettava, etenkin päivitystiedostojen koon ollessa suuri. Yhteyden katkeamisen tai liiallisen hidastumisen seurauksena päivitys voi epäonnistua ja se pitää aloittaa uudestaan. Tällaisessa tilanteessa koko laitteesta voi tulla tilapäisesti käyttökelvoton, jolloin sen palauttaminen käyttökuntoon vaatii erikoistoimenpiteitä. Pahimmillaan laite voi myös vaurioitua pysyvästi korjaamattomaksi. Laiteesta, jonka ohjelmisto on mennyt korjaamattomasti jumiin, käytetään ilmausta brick (tiili).

Opinnäytetyön tavoitteena on kartoittaa eri vaihtoehtoja firmware-etäpäivitysten toteuttamiseksi ja suositella tilaajalle uutta menetelmää ongelman ratkaisemiseksi. Aihe rajattiin koskemaan vain Samsung-merkkisiä Android-puhelimia. Työssä tulee kartoittaa mahdolliset käytettävät ohjelmistot, jakelumediat ja tietoturvanäkökohdat, kuten firmware-tiedostojen salausmenetelmät. Lisäksi tulee huomioida toteutuksen käyttäjäystävällisyys yrityksen sisällä ja loppuasiakkaalla.

Nykyisessä menetelmässä päivitystiedoston siirtäminen ja päivityksen suorittaminen tapahtuu samanaikaisesti. Merkittävä parannus prosessiin saataisiin, jos päivitystiedosto siirrettäisiin ensin asiakkaalle ja päivitys tapahtuisi vasta sen jälkeen. Haasteina tällaisessa prosessissa on mm. päivitysten salaaminen ja salauksen purkaminen, puhelimen mallin varmistaminen sekä päivitysten suorittaminen eri versioista uusimpaan.

2 PÄIVITYSTYÖKALUT

Ennen päivitysmenetelmien kartoittamista on tärkeää tutustua työkaluihin, joilla firmwaren päivittäminen voidaan suorittaa. Samsung tarjoaa kuluttajille internet-sivuillaan kaksi eri vaihtoehtoa virallisten päivitysten asentamiseksi. Näistä Kies on Windows- tai Mac OS -tietokoneella käytettävä ohjelmisto, jonka avulla päivitykset siirretään puhelimeen USB-kaapelin kautta. Kies sisältää myös muita toimintoja, kuten tiedostojen ja yhteystietojen varmuuskopioinnin. (13.)

FOTA (Firmware Over The Air) on Samsungin päivitysmenetelmä, jossa firmware-päivitys ladataan puhelimeen matkapuhelinverkon tai Wi-Fi-verkon yli (14). Sekä Kies että FOTA tukevat vain virallisia Samsungin julkaisemia päivityksiä eivätkä siten sovellu käytettäväksi kustomoidun firmwaren asentamiseen. Menetelmillä voidaan myös mahdollisesti korvata kustomoitu firmware Samsungin virallisella versiolla, joten asiakkaita kielletään päivittämästä laitteitaan itse.

Odin3 on työkalu, jolla voidaan päivittää Samsung-älypuhelimien firmware-ohjelmistoja (15). Ohjelmistoa levitetään freeware- eli ilmaisohjelmalisenssillä. Odin toimii Windows-ympäristössä ja laitteen firmware-päivitys tapahtuu käynnistämällä puhelin download-moodiin ja kytkemällä se USB-kaapelilla tietokoneeseen. Odin-työkalua on hyödynnetty nykyisessä päivitysmenetelmässä.

Heimdall on avoimen lähdekoodin vastine Odin3-työkalulle. Se on saatavilla Windows-, Linux- ja OS X -käyttöjärjestelmille (16). Heimdall-ohjelmiston käyttöliittymä ja toiminta eroavat jonkin verran esikuvastaan. Toinen avoimen lähdekoodin projekti, Jodin3, on edellä mainittujen työkalujen sekoitus. Se on selaimessa toimiva työkalu, joka käyttää Jodin3-päivityspaketteja ja samannäköistä käyttöliittymää. Taustalla Jodin3 hyödyntää kuitenkin Heimdall-ohjelmiston lähdekoodia. (17.)

Firmware-päivitys on useissa puhelimissa mahdollista asentaa myös ilman tietokonetta, käyttäen puhelimen recovery-toiminnallisuutta. Recovery on laitteen muistissa sijaitseva osio, johon laite voidaan käynnistää. Recovery-tilassa on usein mahdollista suorittaa tiettyjä toimenpiteitä, kuten välimuistin tyhjennys,

varmuuskopioiden ottaminen tai virallisen firmware-paketin asentaminen muistikortilta. (18.)

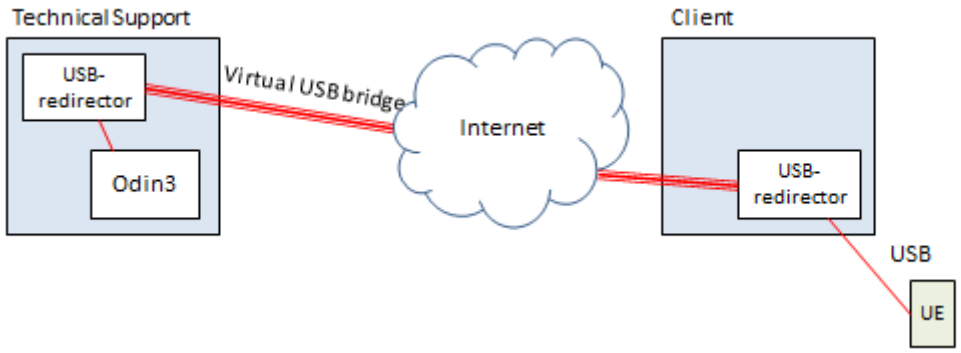
Recovery-tilaan päästään käynnistämällä laite painaen tiettyä näppäinyhdistelmää. Puhelimen alkuperäisen recoveryn lisäksi on laitteeseen mahdollista asentaa jokin kustomoitu recovery, joka tarjoaa yleensä enemmän toimintoja kuin puhelimen alkuperäinen recovery. Näistä mahdollisesti suosituin on ClockworkMod eli CWM. Kustomoitu recovery mahdollistaa usein kustomoidun firmwaren asentamisen. (18.)

3 VAIHTOEHTOISET PÄIVITYSMENETELMÄT

Tässä luvussa on esitetty erilaisia menetelmiä, joilla Anite Finlandin asiakkaiden Samsung-puhelimien firmware-etäpäivitys voitaisiin toteuttaa. Esitettyjä vaihtoehtoja on myös osittain mahdollista yhdistää. Lopulta käyttöön suosittelemani menetelmä voikin olla jokin näiden yhdistelmä.

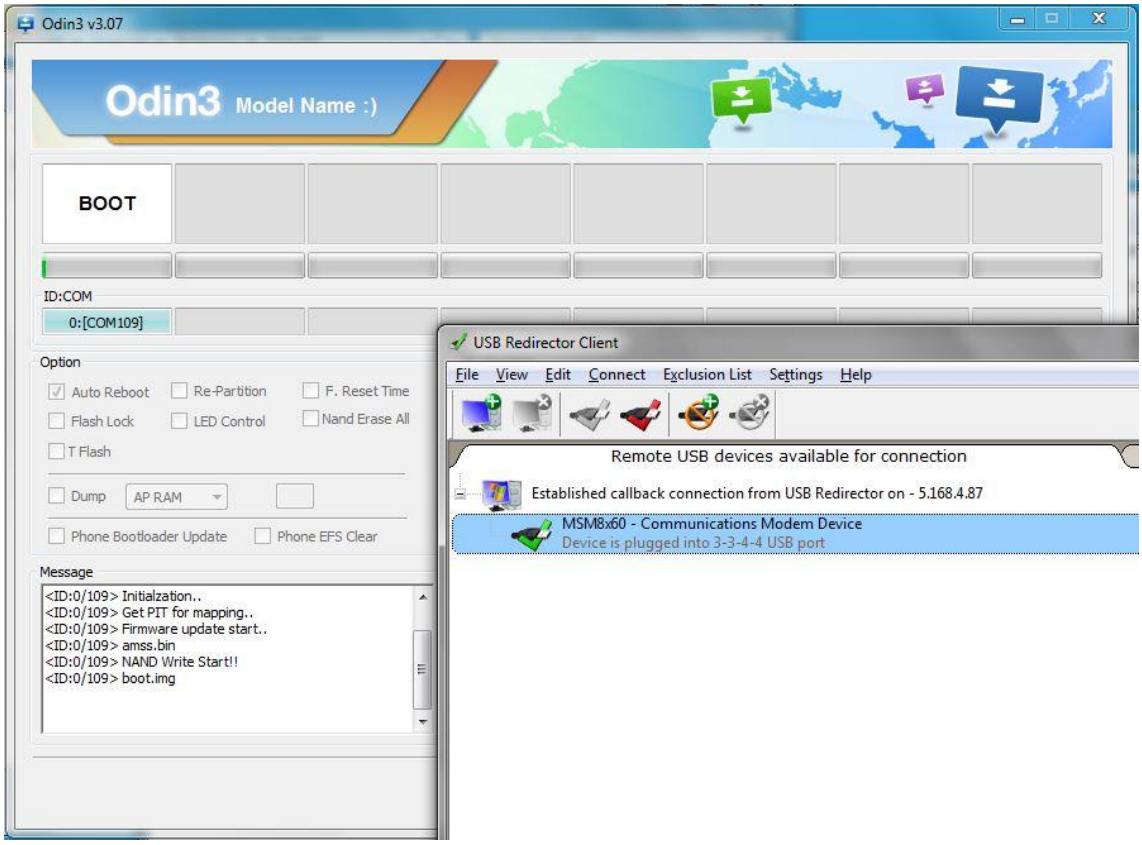
3.1 Nykyinen USB-siltaan pohjautuva menetelmä

Nykyisessä menetelmässä asiakas ottaa yhteyden Nemon tekniseen tukeen (19), jonka kanssa päivityksestä sovitaan. Kun päivitys päätetään toteuttaa, kytkee asiakas puhelimen tietokoneeseensa ja jakaa portin virtuaalisen USB-sillan välityksellä tekniselle tuelle (kuva 1).



KUVA 1. Menetelmän yleiskuva

Virtuaalisen USB-sillan luomiseen käytettävä USB-redirector-ohjelmisto siirtää USB-datan IP-verkon yli (kuva 2). Tekninen tuki suorittaa päivityksen käyttäen Odin-työkalua.



KUVA 2. Aito päivitystilanne, jossa Odin3-ohjelmisto käyttää virtuaalista USB-porttia

Päivityksessä käytetään täyttä järjestelmäkuvaa, joka on kooltaan noin 2 gigatavua. Tiedoston suuren koon vuoksi menetelmä on hidas ja huonolla internet-yhteydellä epävakaa. Myös päivityksen alkuvalmistelut vaativat aikaa. Sujuvan päivitykseen vaaditaan nopea internet-yhteys, jonka vasteajat ovat mahdollisimman lyhyet. Suuri vasteaika yhteydessä voi rikkoa virtuaalisen USB-portin toiminnan.

3.2 Minimipäivitys USB-sillan avulla

Nykyistä menetelmää voisi parantaa siten, että asiakas päivittäisi puhelimeen ensin uusimman virallisen (Samsungin julkaiseman) firmware-ohjelmiston käyttäen Kies- tai FOTA-työkalua. Tämän jälkeen tekninen tuki päivittäisi vain järjestelmän muutosta vaativat osat, kuten käynnistyslataimen.

Tämä vähentäisi USB-sillan läpi siirrettävän datan määrää, jonka vuoksi hitaahko internet-yhteys ei olisi yhtä suuri ongelma kuin nykyisessä menetelmässä. Tämä ei kuitenkaan vähentäisi alkuvalmistelujen määrää teknisen tuen puolelta, sillä päivitystapahtuma pitäisi silti sopia asiakkaan kanssa kuten ennenkin. Menetelmä työllistäisi asiakasta nykyistä menetelmää enemmän ja olisi asiakkaalle sekava.

3.3 Firmwaren salaamaton levitys

Vaihtoehtoinen tapa olisi luovuttaa päivitystiedosto suoraan asiakkaalle tai jälleenmyyjälle, joka suorittaisi päivityksen itse käyttäen Odin3-ohjelmistoa. Jakelukanavana voisi toimia esimerkiksi Nemo User Club palvelu (20), josta teknisen tuen ostaneet asiakkaat voisivat päivitystiedoston ladata.

Menetelmä olisi helppo toteuttaa, mutta vaatisi asiakkaalta hieman enemmän teknistä osaamista nykyiseen menetelmään verrattuna. Tärkeintä asiakkaan kannalta olisi osata valita oikeaan laitemalliin tarkoitettu päivityspaketti. Menetelmä säästäisi teknisen tuen resursseja, sillä asiakas voisi hoitaa päivityksen täysin itsenäisesti. Menetelmä sopisi hyvin myös isomman laitekannan päivittämiseen.

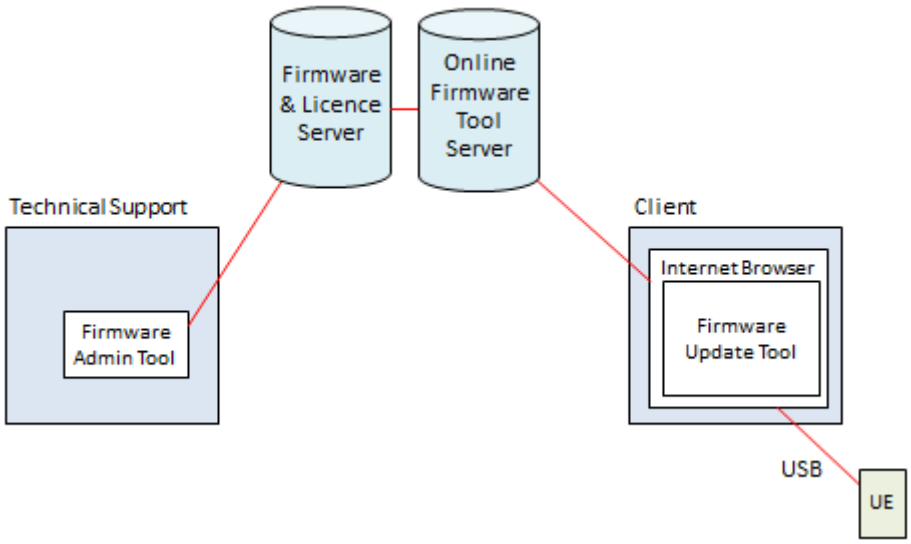
Päivitystiedosto voisi kuitenkin levitä asiakkaille, jotka eivät ole teknisestä tuesta maksaneet, sekä asiakassuhteen ulkopuolisille tahoille. Myös kilpailijat hyötyisivät vapaasti levitettävästä päivityksestä. Tämän menetelmän käyttöönottaminen vaatisi yritykseltä päätöstä, jossa tämä riski hyväksyttäisiin.

Asiakkaan huolimattomuudesta tai ammattitaidon puutteesta voisi myös seurata puhelimen ohjelmiston rikkoutuminen, joka vaatisi laitteen lähettämistä Anitelle korjaustoimenpiteiden suorittamiseksi. Lisäksi tiedostopalvelimella tulisi olla tarvittavasti kaistaa lähettää suuria tiedostoja useille asiakkaille samanaikaisesti.

3.4 Online-työkalu

Asiakkaan kannalta mahdollisesti helpoin menetelmä olisi internet selaimessa toimiva työkalu, jolla päivitys suoritettaisiin. Asiakas kirjautuisi palveluun, kytkisi päivitettävän laitteen tietokoneen USB-porttiin ja aloittaisi päivityksen. Par-

haimmillaan työkalu jopa tunnistaisi asiakaslaitteen mallin ja valitsisi päivitystiedoston automaattisesti. Online-työkalu voisi olla myös alustariippumaton, eli se voisi toimia Windows-ympäristön lisäksi myös Mac/OS X- ja Linux-ympäristössä. Menetelmän yleiskuva on esitelty kuvassa 3.



KUVA 3. Online-työkalua hyödyntävän menetelmän yleiskuva

Ohjelma voisi perustua avoimen lähdekoodin Jodin3-ohjelmistoon. Menetelmän toteuttaminen vaatisi huomattavan paljon ohjelmointia ja testaamista. Menetelmää tarkemmin tutkittaessa pitäisi selvittää, olisiko tiedoston siirtäminen ennen päivityksen suorittamista mahdollista vai olisiko Jodin3 rajattu päivityksen aikaiseen tiedonsiirtoon. Ylläpidettävyyden kannalta järjestelmä olisi hyvin toteutettuna toimiva vaihtoehto.

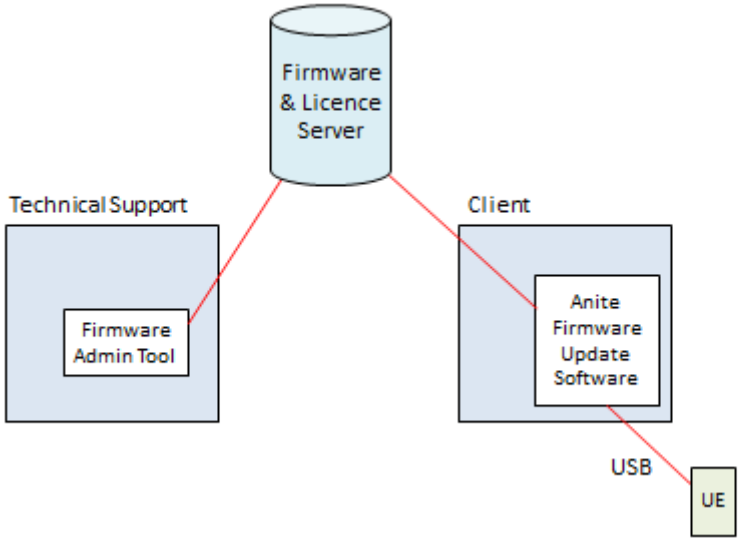
3.5 Salattu päivitystiedosto

Kuten johdannossa todettiin, olisi menetelmään suuri parannus, jos päivitystiedosto siirrettäisiin asiakkaalle ennen päivityksen suorittamista. Myös tiedoston päätyminen ulkopuolisille tahoille haluttaisiin estää. Näiden tavoitteiden saavuttamiseksi olisi vaihtoehtona salata päivitystiedosto jollain salausmenetelmällä, kuten AES.

Salatun tiedoston käyttäminen päivitykseen ei kuitenkaan onnistu suoraan, vaan salaus täytyy purkaa ennen varsinaista päivityksen suorittamista. Salauk-

sen purkaminen käsin ei ole järkevää, sillä tällöin tiedosto on taas altis päätymään salaamattomana kolmansille osapuolille.

Jotta salattu päivitystiedosto voidaan asentaa suoraan laitteeseen, on sitä varten tehtävä erityinen työkalu, joka purkaa salauksen ja suorittaa päivityksen. Päivitykselle tarvitaan myös salaustyökalu. Yleiskuva tällaisen menetelmän rakenteesta on esitetty kuvassa 4. Salattuun päivitystiedostoon perustuvat menetelmät on esitelty seuraavissa alaluvuissa.



KUVA 4. Yleiskuvaus menetelmästä, jossa päivitys on salattu ja sitä hallitaan erityistyökaluilla

3.5.1 Vapaa työkalu

Ensimmäisessä salatun päivitystiedoston menetelmässä salauksen purkava ja päivityksen suorittava ohjelma olisivat vapaassa levityksessä. Ohjelma toimisi kuten Odin, mutta suorittaisi salauksen purkamisen taustalla. Ohjelma olisi myös käyttöliittymältään yksinkertaistetumpi.

Tämä menetelmä soveltuisi hyvin myös useamman laitteen päivitykseen, sillä yhdellä päivitystiedostolla voisi päivittää useamman samanmallisen laitteen. Ilman lisäsuojausta päivitykset voivat päätyä ulkopuolisille tahoille, jos he saavat käyttöönsä myös päivitystyökalun. Tietoturvaltaan tämä ei siis ole salaamatonta päivitystiedostoa parempi ratkaisu eikä täten realistinen vaihtoehto.

3.5.2 Dongle-varmennettu työkalu

Dongle-varmennettu työkalu olisi toteutettu edellisen menetelmän tavoin, mutta työkalun käyttöoikeus varmistettaisiin USB-avaimella (software protection dongle). Työkalun käyttöoikeus luotaisiin ja ylläpidettäisiin vastaavasti kuin yrityksen Nemo Outdoor ja Nemo Analyze -tuotteissa.

Menetelmän käyttöönotto vaatisi ohjelmistokehityksen lisäksi suuren määrän käyttöoikeuslisenssien päivittämistä sekä uusien USB-avaimien lähettämistä asiakkaille, sillä kaikkien Nemo Handy -tuotevariaatioiden mukana ei avainta ole toimitettu.

3.5.3 Recovery-työkalu

Vaihtoehtoinen tapa päivityksen suorittamiselle olisi siirtää päivitystiedosto laitteen muistikortille ja ajaa päivitys laitteen sisäisesti. Tämä menetelmä vaatisi tarkoitusta varten kehitetyn recovery-toiminnallisuuden.

Tässä menetelmässä salaamatonta dataa ei liikutettaisi USB-väylän välityksellä, mikä osaltaan parantaisi tietoturvaa. Menetelmän hankaluutena olisi kuitenkin päivitystiedostojen toimittaminen laitteiden muistikorteille ja ennen kaikkea päivitykseen tarvittava recovery-työkalun asentaminen. Työkalun asentaminen itsessään vaatisi firmware-päivityksen laitteelle, joten menetelmän käyttöönotto jo toimitettujen laitteiden päivitykseen olisi erittäin työlästä.

3.5.4 IMEI-kohtainen päivitys

Edellä mainittuihin menetelmiin voitaisiin ottaa lisävarmistukseksi yksittäiselle laitteelle kohdistetut päivitystiedostot. Päivitystiedosto sisältäisi tunnistetietoja, kuten laitteen IMEI-koodin. Päivityksen tunnistetietoja verrattaisiin laitteen tietoihin ennen päivityksen suorittamista.

Edellä mainittujen menetelmien tavoin myös tätä menetelmää varten tulisi kehittää uusia työkaluja. Lisäksi pitäisi kehittää järjestelmä, jolla yksittäiselle laitteelle kohdennetut päivitystiedostot luotaisiin.

Menetelmällä pyrittäisiin poistamaan riskiä päivityksen joutumisesta ulkopuolisille tahoille. Päivityksen etuna olisi myös se, että päivitystiedosto olisi aina oikealle puhelinmallille tarkoitettu, mikä vähentäisi laiterikon mahdollisuutta. Useamman puhelimen päivitystä varten jouduttaisiin generoimaan kullekin oma päivitystiedosto, mikä lisää työn ja siirrettävän datan määrää.

3.5.5 Online-varmennettu työkalu

Viimeinen vaihtoehtoinen menetelmä olisi luoda kaksi työkalua, joista toinen toimitettaisiin asiakkaille ja toinen olisi järjestelmän ylläpitoa varten. Päivitysyrityksen yhteydessä asiakkaan työkalu tarkistaa päivitysoikeuden palvelimelta. Jos asiakkaalla ja päivitettävällä laitteella on päivitysoikeus, palvelimelta ladataan salatussa muodossa oleva päivitystiedosto ja sen salausavain. Salattu päivitystiedosto tallennetaan asiakkaan tietokoneen kiintolevylle. Tiedoston salaus puretaan päivitystapahtuman aikana.

Asiakkaan käyttöliittymä olisi mahdollisimman yksinkertainen virhemahdollisuuksien minimoimiseksi. Ohjelmisto tunnistaisi kytketyn laitteen ja automaattisesti varmistaisi päivitysoikeuden, lataisi ja suorittaisi päivityksen. Kaikista päivitystapahtumista kirjattaisiin lokia palvelimelle. Ylläpidon käyttöliittymästä olisi mahdollista hallita palvelimella olevia päivityksiä ja seurata statistiikkaa. Menetelmä vaatisi paljon kehitystyötä, mutta parantaisi päivitystapahtumaa joka osa-alueella.

4 TIETOTURVA PÄIVITYKSEN JAKAMISESSA

Menetelmiä kartoittaessa yksi usein esille nousseista käsitteistä oli tietoturva. Käsite on erittäin laaja, mutta tämän työn yhteydessä sillä tarkoitetaan pääasiassa firmware-päivityksen päätymistä yrityksen kilpailijoille tai maksamattomille asiakkaille.

Jokaista tuettua puhelinmallia varten joudutaan kustomoimaan firmware, joka mahdollistaa mittausohjelmiston toiminnan pääkäyttäjän oikeuksin. Tämä työ täytyy myös tehdä aina, kun laitevalmistaja julkaisee laitteelle uudemman firmware-ohjelmiston. Myös kilpailevia ohjelmistoja kehittävät yritykset joutuvat tekemään vastaavaa työtä. Nopea päivitysten tarjoaminen on siis yritykselle kilpailuetu.

Yrityksen politiikan mukaan ohjelmistopäivityksiä tarjotaan vain asiakkaille, joilla on voimassaoleva ”Anite Software Maintenance and Technical Support Agreement”. Käytän tästä sopimuksesta jatkossa lyhennettä TSA. Päivitysmahdollisuutta voidaan siis käyttää myyntivalttina. Palvelusta maksavan asiakkaan tulee voida olettaa, että tuotteen päivitys on helppoa. Tällöin päivitysprosessi pitäisi olla hallussa.

Kun flashays suoritetaan USB-protokollan välityksellä, liikkuu tietokoneen ja päivitettävän laitteen välillä tietoa, jota voidaan haistella erilaisilla USB-sniffer -työkaluilla. Näin nauhoitettua tietoa voidaan analysoida ja sen avulla voidaan kopioida päivityksen sisältö. USB-datan kaappaamiseen ja analysoimiseen käytettävät työkalut voidaan toteuttaa ohjelmallisesti tai laitteistolla. Tämäntyyliseltä hyökkäykseltä on mahdollista suojautua ainoastaan suorittamalla päivitys laitteen sisäiseltä muistikortilta. (21.)

Toinen hyökkäystapa on ottaa päivitetystä puhelimesta niin sanottu system dump. Tällöin laitteen kaikki tiedot kopioidaan ja kopiota voidaan hyödyntää toisen laitteen ohjelmoinnissa. Tällaista hyökkäystä vastaan on lähes mahdoton suojautua, ellei puhelin tue laitesalausta ja firmware kehitettäisiin yhdessä laitevalmistajan kanssa. (22.)

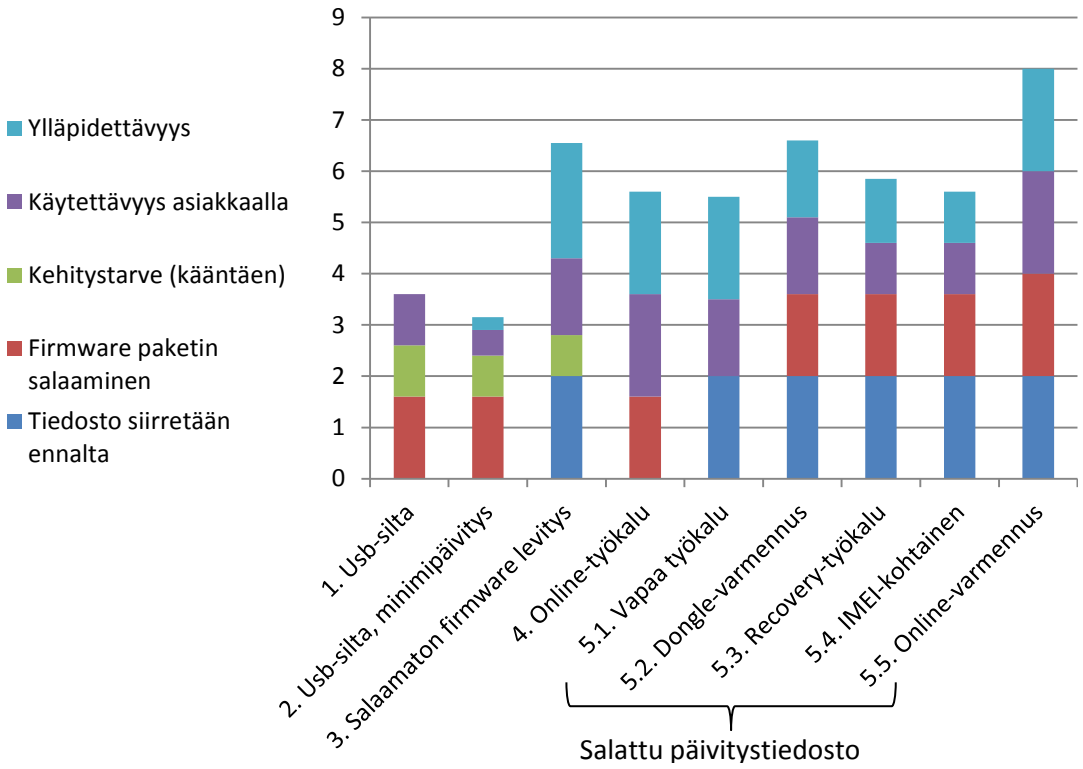
Uuden menetelmän kriteeriksi valittiin, ettei sen tietoturva saa huonontua nykyiseen menetelmään verrattuna. Tavoitteena ei siis ollut kehittää täysin murtamatonta järjestelmää, sillä sen tavoittelemiseen käytettävät resurssit eivät olisi suhteessa salauksen tärkeyteen. Päivitysmenetelmässä käytettävän suojauksen tärkein tehtävä onkin vaikeuttaa päivitystiedoston kaappaamista niin, että hyökkääjän olisi taloudellisempaa keskittyä oman firmware-version kehittämiseen.

5 PÄIVITYSMENETELMIEN ARVIOINTI

Suositeltavan päivitysmenetelmän löytämiseksi oli vaihtoehtoja tutkittava eri näkökulmien kautta. Tärkeimmiksi valintakriteereiksi päätettiin menetelmän toimivuus ja helppokäyttöisyys asiakkaalla, menetelmän ylläpidettävyys sekä tietoturva, joka ei saa huonontua nykyiseen menetelmään verrattuna.

Erittäin tärkeäksi katsottiin myös firmware-päivityspaketin siirtäminen asiakkaalle ennen päivityksen suorittamista. Tällä on suora vaikutus myös menetelmän yleiseen toimivuuteen. Päivitysmenetelmän käytännön tarpeen vuoksi sen kehityskustannukset eivät ole olennaisin valintakriteeri.

Aloitin menetelmien arvioinnin pisteyttämällä ne edellä mainittujen kriteerien perusteella. Pisteytin menetelmät asteikolla 0–10 ja annoin painoarvon jokaiselle kriteerille. Valmis vertailutaulukko ja sen arviointiperusteet löytyvät liitteestä 1. Taulukko laskee painotetut pisteet ja antaa jokaiselle menetelmälle kokonaispisteet samaa asteikkoa käyttäen. Suurempi pistemäärä vastaa parempaa tulosta. Kuvassa 5 on esitetty taulukon tulokset pylväskaaviolla.



KUVA 5. Menetelmien arviointia

Vaikka taulukon pisteytys on toteutettu suuntaa antavasti, voidaan taulukosta helposti havaita, kuinka nykyisen menetelmän puutteet laskevat sen pisteitä merkittävästi. Puutteita löytyy käytettävyydessä, ylläpidettävyydessä ja päivityksen aikaisessa tiedonsiirrossa.

Vaihtoehtoisista menetelmistä päivityksen salaamaton levitys sai taulukossa hyvin pisteitä ja oli helpon toteutettavuutensa vuoksi houkutteleva vaihtoehto. Menetelmässä päivitystiedostojen päätyminen kilpailijoille ja maksamattomille asiakkaille olisi kuitenkin mahdollista. Otin asian esille palaverissa, jonka järjestin ennen lopullista menetelmän valintaa. Asiaa selviteltiin ja tehtiin päätös, ettei päivitystiedostoja haluta levittää vapaasti. Päätökseen vaikuttavia tekijöitä on käsitelty edeltävässä tietoturva-luvussa. Myös Odin3-ohjelmiston levittäminen voisi lisenssisyistä olla hankalaa.

Eri vaihtoehtoisten menetelmien etuja ja heikkouksia miettiessä lähes kaikista vaihtoehdoista löytyi yksi tai useampi syy, jonka vuoksi kyseinen menetelmä hylättiin. Päätökset eri vaihtoehtojen välillä tein lopulta yrityksen asiantuntijoiden kanssa käytyihin keskustelujen pohjalta, eikä edellä esittelemälläni taulukolla

ollut lopputulokseen suoraa vaikutusta. Tulostaulukko on kuitenkin vertailumenetelmänä toimiva, sillä se perustuu arviointikriteeritaulukkoon.

Online-työkalun käyttäminen vaatisi asiakkaalta Java Runtime Environment (JRE) -ohjelmiston asentamista tietokoneelleen. Java mielletään usein tietoturvaltaan riskialttiiksi ympäristöksi ja useissa yrityksissä Java-liitännäisen asentaminen yrityksen tietokoneiden internet-selaimiin onkin kielletty. Tämän lisäksi menetelmä olisi vaatinut huomattavan paljon kehitystyötä.

Salatun päivityksen purkamiseen käytettävistä tekniikoista vapaasti levitettävä työkalu oli vain teoreettinen ratkaisu. Se vastaisi tietoturvaltaan vapaasti jaettavan salaamattoman päivityksen tasoa.

Vakavammin harkittava mahdollisuus oli suojata päivitystyökalu USB-avaimella. Tämä suojaus voitaisiin yhdistää Nemo Outdoor- ja Nemo Analyze -tuotteiden ja joidenkin Nemo Handy -tuotevariaatioiden mukana toimitettavaan kopiosuojaukseen. Tämän menetelmän käyttöönotto sisältäisi kuitenkin logistisia haasteita, sillä kaikilla asiakkailla ei tarvittavaa kopiosuojausavainta ole hallussaan. Lisäksi suojausmenetelmää vastaan on yleisyytensä vuoksi kehitetty murtamistapoja.

Eri lähestymistapa firmware-päivitykseen on suorittaa päivitys laitteen sisäiseltä muistikortilta, jolloin laitetta ei tarvitsisi liittää tietokoneeseen päivityksen ajaksi. Tämä ratkaisu poistaisi USB-väylään kohdistuvat hyökkäykset. Menetelmä vaatisi kuitenkin puhelimeen kustomoidun recovery-ohjelmiston, jonka asentaminen jo asiakkailla oleviin puhelimiin on menetelmän suurin kompastuskivi. Myös päivitysten toimittaminen asiakkaalle olisi iso haaste.

Päivitykset on mahdollista luoda IMEI-kohtaisiksi, jolloin jokaiselle päivitettävälle laitteelle luotaisiin oma päivitystiedostonsa. Tällainen järjestelmä olisi kuitenkin työläs ylläpidettävä ja vaatisi erittäin paljon datan siirtoa. Menetelmä soveltuisi erittäin huonosti suuremman laite-erän päivittämiseen.

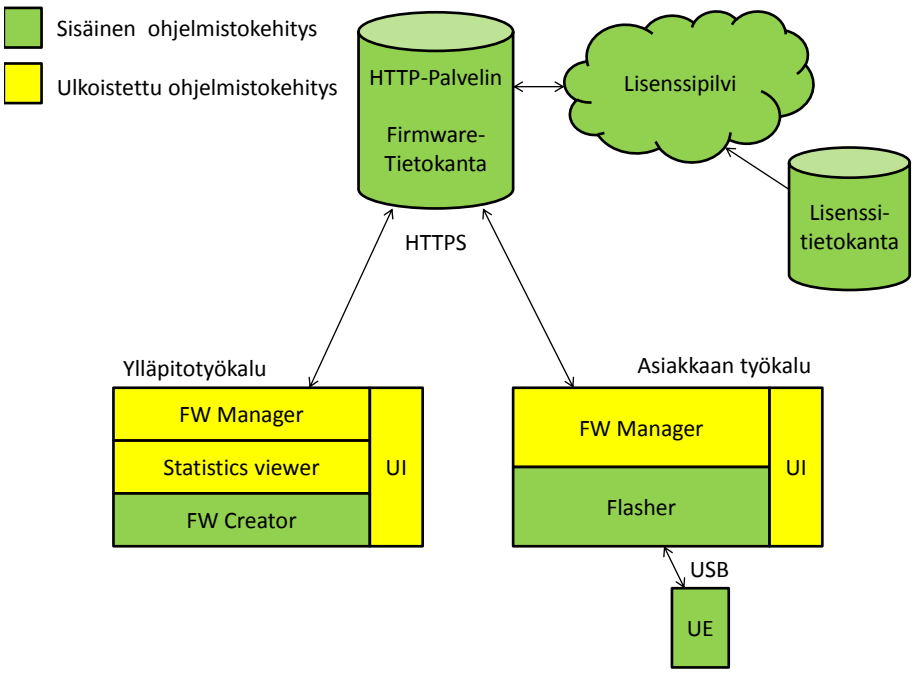
Päivitystyökalun käyttöoikeus voitaisiin myös tarkistaa lisenssitietokannasta jokaisen päivityksen yhteydessä. Menetelmän asiakas- tai laitekohtainen päivitysoikeus olisi hyvä ominaisuus etenkin kun se mahdollistaisi saman päivitystie-

doston käyttämisen useamman samanmallisen laitteen yhtäaikaiseen päivittämiseen. Tämä vähentäisi merkittävästi siirrettävän datan määrää. Tätä menetelmää päätettiin tutkia tarkemmin ja se on esitelty laajemmin seuraavassa luvussa.

6 VALITTU MENETELMÄ

Päädyin lopulta ehdottamaan menetelmää, jota varten kehitettäisiin kaksi erillistä työkalua. Toinen työkaluista luovutettaisiin asiakkaiden käyttöön ja toinen olisi tarkoitettu yrityksen sisäiseen käyttöön järjestelmän ylläpitoa varten. Järjestelmä vaatisi myös tietokantoja ja palvelimia, joissa sijaitsisivat päivitystiedostojen lisäksi tiedot asiakkaista, heidän ostamistaan lisensseistä ja päivitysoikeuksista. Tietokantoihin kerättäisiin myös statistiikkaa päivitysten suorittamisista.

Tilaaja hyväksyi ehdotuksen ja järjestelmän määrittelemistä päätettiin jatkaa yhdessä alihankkijan kanssa. Lopullisen päätöksen järjestelmän kehittämisestä tilaajayritys tekee, kun määrittelyä on tarkennettu ja alihankkija on tehnyt siitä tarjouksen. Järjestelmän kehityksessä päätettiin hyödyntää alihankkijaa kuvan 6 osoittamalla tavalla.



KUVA 6. Järjestelmän kokonaiskuva

Uuden järjestelmän kehittämisessä yrityksen sisäiseen kehitykseen jäisi firmware-päivitysten luomiseen ja suorittamiseen käytettävät ohjelmisto-osat sekä palvelimille toteutettavat tietokannat ja rajapinnat. Kuvaan keltaisella merkitty ali-

hankkijalla teetettävä osuus on periaatteessa melko yleismallinen tiedonsiirto-ohjelmisto ja sen käyttöliittymä.

6.1 Firmware-päivitystiedoston luominen ja ylläpitotyökalu

Päivityspaketti luodaan ylläpitotyökalun FW Creator -osaa hyödyntäen. Päivityspaketti pitää sisällään varsinaisen laitemallikohtaisen firmware-päivityksen ja ohjeet päivityksen suorittamiseksi. Nämä ohjeet esitetään asiakkaan työkalun käyttöliittymässä päivitystapahtuman aikana.

Ylläpitotyökalun käyttöliittymän kautta uusi päivityspaketti siirretään palvelimelle ja sille määritellään käyttöoikeudet. Ensimmäisessä vaiheessa uusi päivitys voi olla testivaiheessa, jolloin se on saatavilla vain yrityksen sisäisesti sekä valituilla asiakkailla. Kun uutta päivitystä on testattu tarpeeksi, sen jakelua voidaan laajentaa. Vanhemmat päivitysversiot pidetään tietokannassa, mutta niiden lataaminen vaatii asiakkaalta erityistoimenpiteitä. Tiettyjä päivityksiä voi olla myös asiakkaiden käytössä, vaikkei heillä voimassaolevaa TSA-sopimusta olisikaan.

Käyttöliittymässä on näkymät myös statistiikalle ja mahdollisesti käyttäjien käyttöoikeuksien hallinnalle. Statistiikkaa varten on mahdollista määrittää parametreja, joiden perusteella tietokantahaut suoritetaan. Tulokset esitetään taulukkomuodossa, mutta myös karttanäkymä voisi olla mahdollinen.

6.2 Firmware-päivitys asiakkaalla ja asiakkaan työkalu

Asiakas lataa tietokoneelleen päivitystyökalun ja kytkee päivitettävän laitteen USB-kaapelilla. Ohjelmisto hakee laitteelta tunnistetietoja käyttäen ADB-komentoja. Näitä tunnistetietoja hyödyntäen se tarkistaa palvelimelta laitteelle saatavilla olevat päivitykset. Statistiikkaa varten kerätään tietoa asiakkaan sijainnista myös IP-osoitteen perusteella. Jos päivityksiä on saatavilla useampi, on niistä yksi merkitty oletukseksi. Muun kuin oletusvaihtoehdon valitseminen edellyttää vastuunvapautuslausekkeen hyväksymistä.

Päivityspaketti sisältää myös tiedon sen käyttämiseen tarvittavasta ohjelmistoversiosta. Jos asiakkaan työkalu on liian vanha, se lataa palvelimelta uudemman version automaattisesti.

Päivityspaketin lataaminen asiakkaan koneelle tapahtuu taustalla, kun käyttäjää samalla informoidaan edistymispalkilla. Kun tiedosto on ladattu, sen sisältämät päivitysohjeet esitetään käyttöliittymässä päivityksen edetessä. Päivityspaketin salaus puretaan flashayksen yhteydessä, eikä päivityspakettia tallenneta salaamattomana tietokoneen muistiin. Päivityksen päätteeksi palvelimelle lähetetään statistiikkatietoja päivityksestä.

Käyttöliittymän tulee olla helppokäyttöinen ja ohjelman tulee suorittaa tarvittavat toimet taustalla käyttäjää tarpeen tullen ohjeistaen. Käyttöliittymässä oleva ohjeistus opastaa asiakasta tekemään vaadittavat toimet, kuten tarvittavien näppäinyhdistelmien painamisen laitteessa. Ohjelma voisi myös esittää arvioita päivityksen kestosta ja ajasta, jonka kuluttua käyttäjän toimia tarvitaan.

6.3 Menetelmän haasteet

Menetelmän kehittämisessä on useita haasteita. Prosessin alussa menetelmä pitää määritellä ja siitä pitää toteuttaa tarjouspyyntö alihankkijalle. Jokainen menetelmän osa-alue vaatii tuotekehitystä ja testaamista. Näitä ovat esimerkiksi ylläpitotyökalu, asiakkaan työkalu, lisenssitietokanta, asiakkaan ja laitteen tunnistus, statistiikan kerääminen ja esittäminen, ohjelmistopäivitykset sekä firmwaren päivittäminen vanhempaan versioon.

Suurien tiedostojen salaaminen ja siirtäminen IP-verkon yli on myös haaste. Menetelmän tulee pystyä jatkamaan tiedoston siirtoa, vaikka yhteyden vasteajat olisivat huonot tai yhteys katkeaisi tiedoston siirron aikana.

6.4 Menetelmän edut

Hyvin toteutettuna uusi menetelmä täyttäisi sille työn alussa asetetut tavoitteet. Sen käytettävyys olisi huomattavasti parempi niin asiakkaalla kuin ylläpidossa. Tiedoston siirtäminen olisi varmempaa ja siten myös päivitystapahtuma luotettavampi.

Menetelmä mahdollistaa statistiikan keräämisen päivitystapahtumista, eikä siinä käytetä ulkopuolisia ohjelmistoja. Myös logistiikka tulee hyödyntämään uusia työkaluja toimitusten yhteydessä.

Uusi menetelmä mahdollistaa myös uusien laitteiden firmware-päivitykset asiakkaan tiloissa. Tämän ansiosta asiakas voi ostaa laitteen valitsemaltaan toimittajalta, ilmoittaa sen tiedot Anitelle ja suorittaa käyttöönoton itsenäisesti.

7 OHJELMISTON MÄÄRITTELEMINEN

Uuden järjestelmän kehittämisen tueksi tulee ohjelmisto määritellä hyvin kehitystyön alkuvaiheessa. Erityisen tärkeää tämä on, kun kehitystyössä käytetään alihankkijaa. Aikataulun niin salliessa päätimme laajentaa opinnäytetyön koskemaan myös ohjelmistomäärittelyn ensimmäistä vaihetta, jossa alihankkijalle laaditaan alustava määrittely.

Menetelmän valinnan jälkeen järjestin pari palaveria, joissa kävimme läpi järjestelmän toimintaa, sen eri rajapintoja ja hyödynnettäviä tekniikoita. Palavereihin osallistui yrityksen asiantuntijoita, jotka tulevat myös osaltaan kehittämään tulevaa järjestelmää.

Palavereissa päätettiin, mitkä osiot kokonaisuudesta tullaan kehittämään itse ja mitkä osat teetetään alihankintana. Päätimme myös, että tiedonsiirtoon työkalujen ja palvelimen välillä käytetään HTTPS-protokollaa. Tämä protokolla katsottiin tietoturvan ja yleisen toiminnan kannalta järkeväksi ratkaisuksi. Useimpien yritysten palomuurit päästävät HTTPS-liikenteen läpi, kun taas muiden protokollien portteja voi olla turvallisuussyistä suljettu.

Alihankkijaa lähestyttiin määrittelydokumentilla, jossa järjestelmä kuvattiin yleisellä tasolla. Dokumentti löytyy liitteestä 2. Määrittelydokumentin tehtävänä oli saada alihankkijalta alustava hinta-arvio järjestelmän kehityskustannuksista ja tarkentavia kysymyksiä sen toiminnasta.

Järjestelmän tarkempi määrittely tehdään alihankkijan kanssa yhteistyönä. Määrittelyssä sovitaan järjestelmän kehityksen aikataulu, tarkat tekniset vaatimukset sekä mahdolliset jatkossa lisättävät ominaisuudet.

8 YHTEENVETO

Työn päätarkoituksena oli kartoittaa vaihtoehtoisia menetelmiä yrityksen asiakkailla olevien Android-puhelimien firmware-päivityksiä varten. Kartoitetuista vaihtoehdoista piti valita paras ja esitellä se tilaajalle. Työn edetessä opinnäytetyö laajennettiin koskemaan myös uuden menetelmän vaatimusmäärittelyn ensimmäistä vaihetta, jossa alihankkijaa lähestyttiin tarjouspyynnöllä.

Tuloksena kartoitin kahdeksan vaihtoehtoista menetelmää etäpäivityksen suorittamiseksi. Valitsin näistä mielestäni parhaan ja esittelin sen yhdessä hylättyjen vaihtoehtojen kanssa tilaajalle. Tilaaja hyväksyi menetelmän alustavasti. Lopullisen päätöksen uuden menetelmän kehittämisestä ja käyttöönottamisesta tilaaja tekee saatuaan tarjouksen alihankkijalta.

Opin työn edetessä paljon uusia asioita. Jouduin aktiivisesti hakemaan tietoa mm. Android-laitteisiin, tietoturvaan, salausmenetelmiin, IP-verkkoihin, tietokantoihin ja ohjelmointiin liittyvissä asioissa. Itsenäisen tiedonhaun lisäksi pystyin myös hyödyntämään ohjaavan opettajan ja tilaajayrityksen ammattilaisten tietotaitoa. Yrityksessä järjestämäni palaverit olivatkin työn etenemisen kannalta merkittäviä tapahtumia.

Varsinaisia ongelmia ei opinnäytetyöprosessin aikana esiintynyt. Työssä piti tutkia päivitysmenetelmiä monelta eri kantilta, joten oman osaamisen jatkuva kehittäminen ammattimaisten valintojen tekemiseksi oli itselleni suuri haaste.

Olen tyytyväinen työn lopputulokseen. Onnistuin kartoittamaan useita vaihtoehtoisia menetelmiä ja valitsemaan niistä tilaajalle sopivimman. Valinnassa käytin hyväkseni itsenäisellä tiedonhaulla oppimiani asioita, ohjaavan opettajan näkemyksiä ja kollegoiden mielipiteitä. Työn tulos vastaa hyvin alkuvaiheessa asetettuja tavoitteita.

Uuden etäpäivitysmenetelmän kehittämistyö on kokonaisuudessaan paljon laajempi kuin tähän työhön rajattu osuus. Lopullinen käyttöön otettava menetelmä voi poiketa tässä raportissa esitellystä menetelmästä suurestikin. Menetelmään on myös jo kaavailtu useita lisätoimintoja, joita mahdollisesti lisätään tulevai-

suudessa. Tulen olemaan uuden menetelmän kehitysprosessissa mukana myös opinnäytetyön ulkopuolelle rajatuilta osin.

LÄHTEET

1. Karvonen, Tuomas 2006. Anite ostaa Elektrobitin tyttäriä. It-viikko. Saatavissa: http://www.itviikko.fi/talous/2006/11/03/anite-ostaa-elektrobitin-tyttaria/200619642/7. Hakupäivä 29.1.2014.

2. Products. 2014. Anite. Saatavissa: http://www.anite.com/businesses/network-testing/products#.UtvbvZX8JhE. Hakupäivä 29.1.2014.

3. Elgin, Ben 2005. Google Buys Android for Its Mobile Arsenal. WebCite. Saatavissa: http://www.webcitation.org/5wk7sIvVb. Hakupäivä 29.1.2014.

4. Etherington, Darrell 2013. Android Nears 80% Market Share In Global Smartphone Shipments, As iOS And BlackBerry Share Slides, Per IDC. TechCrunch. Saatavissa: http://techcrunch.com/2013/08/07/android-nears-80-market-share-in-global-smartphone-shipments-as-ios-and-blackberry-share-slides-per-idc/. Hakupäivä 30.1.2014.

5. Across Fonblets and Phablets Samsung Has 63% Share of all Android Mobile Devices. 2013. Localytics. Saatavissa: http://www.localytics.com/blog/2013/fonblets-and-phablets-samsung-has-share-of-android-mobile-devices/. Hakupäivä 29.1.2014.

6. Jung-a, Song – Oliver, Christian 2010. Samsung beats HP to pole position. Financial Times. Saatavissa: http://www.ft.com/cms/s/2/c48d477a-0c3b-11df-8b81-00144feabdc0.html#axzz2qqmExgSS. Hakupäivä 29.1.2014.

7. Samsung overtakes Nokia in mobile phone shipments. 2012. BBC. Saatavis-sa: http://www.bbc.co.uk/news/business-17865117. Hakupäivä 29.1.2014.

8. Nemo Handy. 2013. Anite. Saatavissa: http://www.anite.com/sites/default/files/Nemo%20Handy%20Brochure%20Oct%202013_0.pdf. Hakupäivä 29.1.2014.

9. Tyler, Lee 2012. Rooting for Android: What, why and how? Ubergizmo. Saatavissa: http://www.ubergizmo.com/2012/05/rooting-for-android-what-why-and-how/. Hakupäivä 29.1.2014.

10. Raja, Haroon 2012. What Is Meant By Firmware, Stock & Custom ROMs And Flashing. AddictiveTips. Saatavissa: http://www.addictivetips.com/mobile/what-is-meant-by-firmware-stock-custom-roms-and-flashing-guide/. Hakupäivä 29.1.2014.

11. Android 101: What Is A Bootloader? 2012. Android-Does. Saatavissa: http://www.android-does.com/about-android-does/. Hakupäivä 30.1.2014.

12. Hildenbrand, Jerry 2012. Android A to Z: What is a kernel? Android Central. Saatavissa: http://www.androidcentral.com/android-z-what-kernel. Hakupäivä 30.1.2014.

13. Ohjelmistopäivitys – Kiesin kautta. 2014. Samsung. Saatavissa: http://www.samsung.com/fi/support/usefulsoftware/SWUP/JSP. Hakupäivä 21.3.2014.

14. Ohjelmistopäivitys – Fotan kautta. 2014. Samsung. Saatavissa: http://www.samsung.com/fi/support/usefulsoftware/FOTA/JSP. Hakupäivä 21.3.2014.

15. Cheng, Felix 2014. Odin3 Review. Lo4d.com. Saatavissa: http://odin3.en.lo4d.com/. Hakupäivä 24.3.2014.

16. Heimdall. 2014. Glass Echidna. Saatavissa: http://glassechidna.com.au/heimdall/. Hakupäivä 24.3.2014.

17. Outler, Adam 2014. Jodin3, web browser or offline flashing tool. CASUAL-Dev. Saatavissa: http://casual-dev.com/2014/01/04/jodin3-web-browser-or-offline-flashing-tool/. Hakupäivä 24.3.2014.

18. Raja, Haroon 2011. What Is ClockworkMod Recovery And How To Use It On Android [Complete Guide]. AddictiveTips. Saatavissa:

http://www.addictivetips.com/mobile/what-is-clockworkmod-recovery-and-how-to-use-it-on-android-complete-guide/. Hakupäivä 24.3.2014.

19. Nemo Techical Support Contacts. 2014. Anite. Saatavissa: http://www.nemotechnologies.com/index.php?312. Hakupäivä 13.2.2014.

20. Nemo User Club. 2014. Anite. Saatavissa: http://www.nemotechnologies.com/index.php?241. Hakupäivä 13.2.2014

Benefits of Using a Hardware USB Protocol Analyzer over a Software-Only Solution. 2013. Total Phase. Saatavissa: http://www.totalphase.com/solutions/apps/usb-analyzer-benefits/. Hakupäivä 19.3.2014.

21. Samantha 2013. Create Your Own System Dump Files from Android Firmwares. XDA Developers. Saatavissa: http://www.xda-developers.com/android/create-your-own-system-dump-files-from-android-firmwares/ . Hakupäivä 19.3.2014.

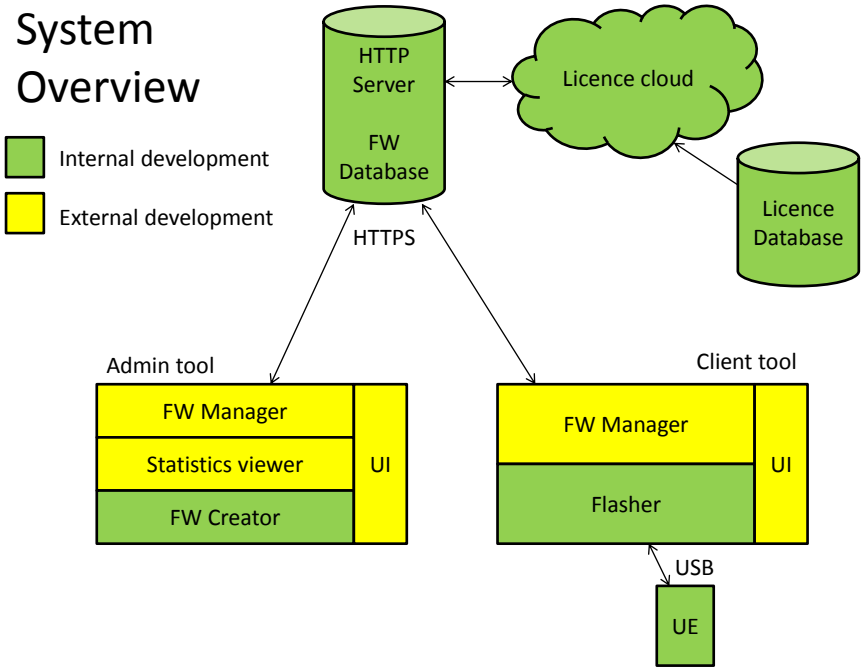
	0	2	4	6	8	10
Tiedosto siirretään ennalta	Päivitystiedosto siirretään päivityksen aikana					Päivitystiedosto siirretään ennalta
Firmware paketin salaaminen	Päivitys vapaasti saatavilla	Päivitys vaatii heikon tunnistuksen			Jonkinlainen käyttöoikeuden tarkistus. Päivitys kryptattu.	Päivitysoikeus tarkistetaan joka kerta usealla parametrilla. Päivitys kryptattu.
Kehitystarve (kääntäen)	Vaatii paljon ohjelmointia ja testausta	Vaatii huomattavan määrän ohjelmointia ja testausta			Vaatii muutosta päivitysprosessiin	Järjestelmä käytössä
Käytettävyys asiakkaalla	Vaatii asiakkaalta ammattitaitoa. Monimutkainen menetelmä		Vaatii joko teknistä osaamista tai epäselvä menetelmä		Ei vaadi ammattitaitoa asiakkaalta. Yksinkertainen menetelmä.	Täysin automaattinen menetelmä
Ylläpidettävyys	Ylläpito vaatii paljon ylläpidon resursseja jokaiseen päivitystapahtumaan		Ylläpito vaatii päivityskohtaista työtä	Ylläpito vaatii asiakaskohtaista työtä	Vaatii tietokannan ylläpitämistä	Vaatii ainoastaan firmware-tietokannan ylläpitämistä

Kriteeri	Painoarvo	1. Usb-silta		2. Usb-silta, minimipäivitys		3. Salaamaton firmware levitys		4. Online-työkalu		5.1. Vapaa työkalu		5.2. Dongle-varmennus		5.3. Recovery-työkalu		5.4. IMEI-kohtainen		5.5. Online-varmennus	
Tiedosto siirretään ennalta	20 %	0	0	0	0	10	2	0	0	10	2	10	2	10	2	10	2	10	2
Firmware paketin salaaminen	20 %	8	1,6	8	1,6	0	0	8	1,6	0	0	8	1,6	8	1,6	8	1,6	10	2
Kehitystarve (kääntäen)	10 %	10	1	8	0,8	8	0,8	0	0	0	0	0	0	0	0	0	0	0	0
Käytettävyys asiakkaalla	25 %	4	1	2	0,5	6	1,5	8	2	6	1,5	6	1,5	4	1	4	1	8	2
Ylläpidettävyys	25 %	0	0	1	0,3	9	2,3	8	2	8	2	6	1,5	5	1,3	4	1	8	2
Yhteensä	100 %	22	**3,6**	19	**3,2**	33	**6,6**	24	**5,6**	24	**5,5**	30	**6,6**	27	**5,9**	26	**5,6**	36	**8,0**

Firmware remote update system

Specification v.1.0

System Overview



- The system consists of three main parts: admin tool, client tool and the cloud
- The admin tool and the client tool are different builds of partly the same software
- FW Manager is part of the tools that handles communication and file transfers between Flasher/FW Creator and the cloud
- Statistics viewer is part of the admin tool that plots information of update events in table and map forms
- HTTPS protocol is used for file transfers and communication between the server and the tools

FW Update Process in General

- Firmware Update package is created
  - Flash information is generated
  - Flash instructions are generated
- FW package is created for testing.
  - In the testing phase the distribution of the package is limited

    - If there is problems those are fixed and FW package version info is increased
- FW package is released when tests are passed
- When FW package life cycle ends it is archived
    - It might be available to clients in certain situations

- User connects the device to the PC/laptop
- Client tool detects the device, requires available updates from the server
- Option of available updates are given to the user
- Selected update is downloaded to local storage (If the file is already downloaded, the local file is used)
- Client tool update is downloaded and installed if necessary for the update
- Firmware update is executed
- Statistics about the update is sent to the server

Device FW Update Session

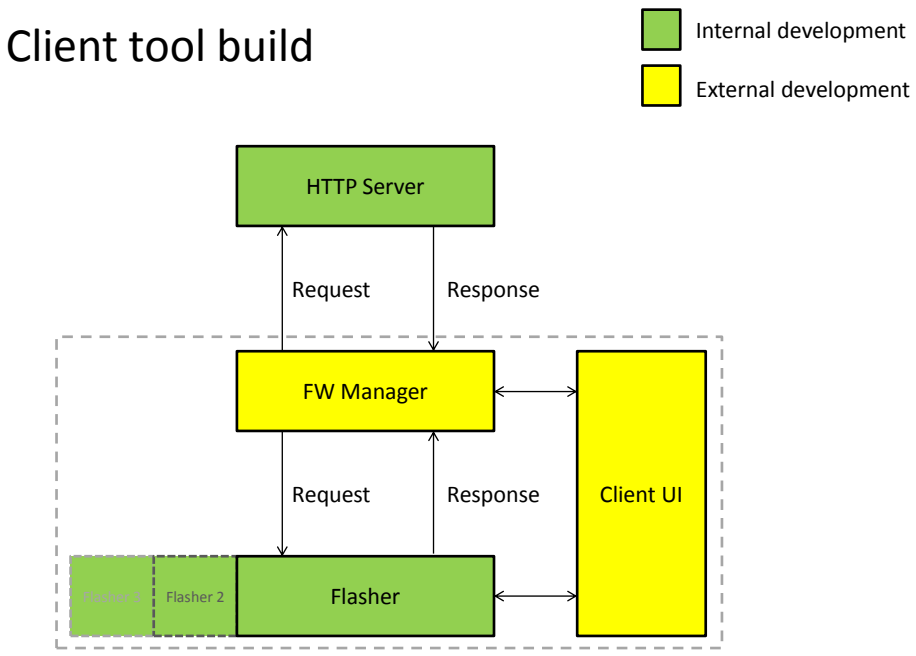
The description of the FW update session of one device.

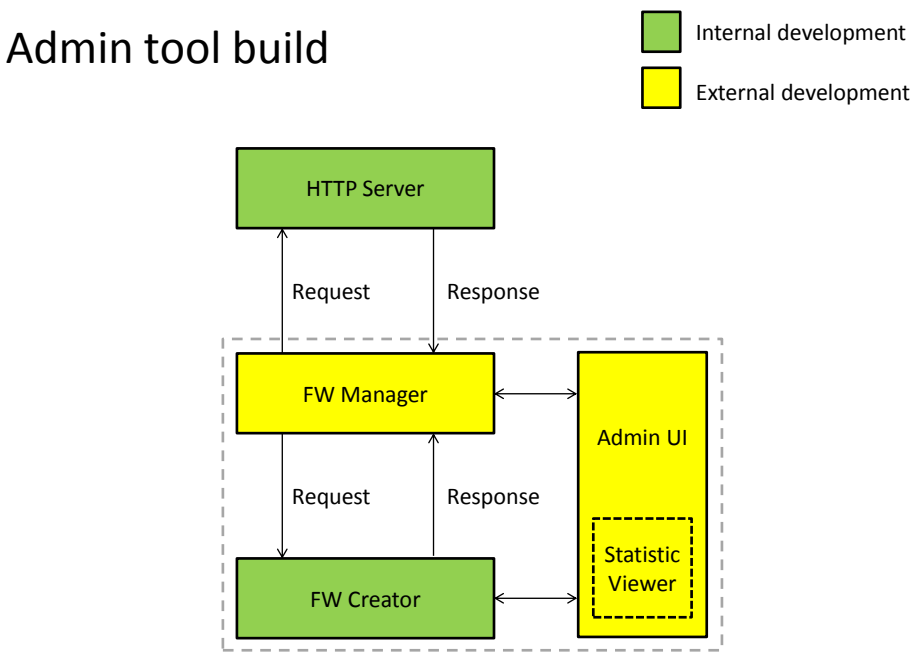
1. User connects the device by USB cable.
2. Flasher sends request for flash to FW Manager. Request includes device information (IMEI, product type, device model)
3. FW Manager requests list of available updates from the server for the device
4. Server replies with OK or FAIL. FAIL is returned if IMEI is unauthorized or blacklisted. OK response includes list of available software versions.
5. User is informed if the device is not allowed to update. “Please contact Nemo support”.
6. If the server returns available updates, user is asked to continue the update. For special cases if older FW version is needed, user can request for older firmware by agreeing a disclaimer.
7. FW Manager searches local storage for the update.
8. Request about selected update is sent to the server.
9. Server replies with URL for update file download, information about SW version to be used, URL for downloading latest SW, MD5 checksum for the update, MD5 checksum for the SW installer and the decryption key for the FW update file
10. If the software version required for the update is newer than the one in use, the newer version is downloaded from the given URL, compared to

MD5 checksum and the tool is updated. If newer SW version is available but not necessary, user is asked to update the tool.

11. If the update was found on local storage, it will be compared to the MD5 checksum. If it matches it will be used. Otherwise the update is downloaded from the URL given by the server.
12. The decryption key and the URL for local firmware update file are passed to the flasher
13. Flasher returns OK if the file is found and decrypted successfully
14. Flasher returns OK/FAIL after the update.
    Statistics about the update is sent to the server

There might be several FW update sessions simultaneously.





FW Manager

- Same SW for both client and admin tools
- Communicates with Flasher, HTTP Server and User Interface.
- Main task is to handle file downloads.
- Resume capability for file transfers.
- Communication with Flasher/FW creator, Server and User

Client UI

- Very simple and easy to use
- Should include a push button for starting the firmware upgrade, a progress bar for indicating the progress of file download and flashing, text and picture notifications for user instruction
- User instructions are included in the downloaded firmware update package

Administrator UI

- Requests list of available files on the HTTP server via FW Manager. Prints file list to UI
- Controls for uploading and removing files and client software installers on the server
    - Upload parameters: Key, MD5, Product, Device, Version
- Controls for setting info about files on the server
    - Sortable by columns
- Controls for setting permission for each user and device to allow/deny updates

Statistic Viewer

- Part of Administrator UI
- Requires information from the server
    - IMEI, date, image name, product, device, version, IP address, download count, successful flash count..
    - Sortable by colums
- Plots information to a list and a map
- Controls for setting time span

HTTP Server

- HTTP Server has a database for customer, firmware file, software and statistics information
- It also works as a file server for the FW files and software installers
- These functionalities can be physically on separate Hardware

Data structures

- Firmware Information
    - Device
    - Model
    - Firmware version
    - Flash instructions
    - Distribution (Testing, Current version, Archived)
    - Download URL
    - …
- Flash Session
    - ID (Primary Key)

    - IMEI
    - IP Address
    - …
- Statistics
    - Timestamp
    - FS ID (Foreign Key)
    - Type (Login/Error/Info)
    - …
- …

Database

- Firmware information -table has the information about firmware update packages
- Flash Session -table has the information about updated devices
- Statistics-table has the information about update events. The data is used in Admin tool/Statistics viewer.