Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Návrh a implementace podpůrného nástroje pro distribuci softwaru

Jan Dryml

Bakalářská práce

2020

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Jan Dryml**
Osobní číslo:	**I16076**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Téma práce:	**Návrh a implementace podpůrného nástroje pro distribuci softwaru**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování

Cílem práce je návrh a implementace podpůrného softwarového nástroje pro distribuci nových verzí monitorovacího softwaru na Raspberi Pi v rámci virtuální sítě.

Teoretická část bakalářské práce bude věnována segmentu IoT, virtuálním sítím a popisu technologií, které budou využity v rámci praktické části.

Praktická část práce bude zaměřena na vlastní návrh implementace podpůrného softwarového nástroje pro distribuci nových verzí softwaru rámci virtuální sítě. Součástí práce bude i generování a správa nových certifikátů na vhodném úložišti pro vlastní VPN síť. Rovněž bude nutné pro správných chod aplikace navrhnout vhodný databázový model.

Výsledný systém bude testován v reálném prostředí.

Rozsah pracovní zprávy: **min 30**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

CHIN, Stephen a James L WEAVER. *Raspberry Pi with Java: programming the internet of things (IoT).* New York: McGraw-Hill Eucation, [2016]. ISBN 978-0071842013.

SCHWARTZ, Março. *Building Smart Homes with Raspberry Pi Zero* New York: Packt Publishing, [2016]. ISBN 978-1786466952.

Vedoucí bakalářské práce: **Ing. Jan Fikejz, Ph.D.**
Katedra softwarových technologií

Datum zadání bakalářské práce: **15. listopadu 2019**
Termín odevzdání bakalářské práce: **7. května 2020**

Ing. Zdeněk Němec, Ph.D.
děkan

Ing. Lukáš Čegan, Ph.D.
pověřený vedením katedry

V Pardubicích dne 17. prosince 2019

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 5.5.2020

Jan Dryml

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu práce Ing. Janu Fikejzovi, Ph.D. za vstřícný přístup, jeho čas a rady, které mi věnoval při zpracovávání bakalářské práce.

ANOTACE

Bakalářská práce se zabývá implementací podpůrného softwarového nástroje pro efektivní distribuci nových verzí monitorovacího softwaru, běžících v rámci jednodeskových počítačů s libovolnou distribucí Linuxu. Pro tvorbu grafického rozhraní je využitá platforma JavaFx. Součástí práce je generování a správa nových certifikátů pro vlastní VPN síť. V práci je popsán segment IoT, virtuální privátní sítě a technologie použité v implementaci. Také je zde samotný návrh aplikace a seznámení s funkcemi aplikace. K uchovávání dat je vyžita databáze MySQL.

KLÍČOVÁ SLOVA

Distribuce softwaru, podpůrný nástroj, certifikáty, VPN, Raspberry Pi, JavaFx, IoT, SSH, MySQL

TITLE

Design and implementation of software distribution support tool

ANNOTATION

The bachelor thesis deals with the implementation of a support tool to facilitate the work and distribution of new versions of monitoring software running on single-board computers with arbitrary Linux distribution. For creation of graphical user interface is used JavaFX platform. The work also includes the generation and management of new certificates for own VPN network. The thesis describes the segment of IoT, virtual private networks and technologies used in implementation. There is also the design of the application and acquaintance with the functions of the application. Data are persisted in MySQL database.

KEYWORDS

Software distribution, support tool, certificates, VPN, Raspberry Pi, JavaFX, IoT, SSH, MySQL

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM ZKRATEK

API	Application Programming Interface
DDL	Data Definition Language
DDOS	Distibuted Denial of Service
FTP	File Transfer Protocol
FXML	eFfects eXtended Markup Language
GPS	Global Positioning System
GUI	Graphical User Interface
HQL	Hibernate Query Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment)
IP	Internet Protocol
IoT	Internet of Things
JPA	Java Persistence API
JSON	JavaScript Object Notation
MVC	Model–View–Controller
ORM	Object Relational Mapping
POM	Project Object Model
RDBMS	Relational Database Management System
RSA	Rivest–Shamir–Adleman cryptosystem
SFTP	SSH File Transfer Protocol
SQL	Structured Query Language
SSH	Secure Shell
VPN	Virtual Private Network
Wi-Fi	Wireless Fidelity
WORA	Write Once, Run Anywhere
WWW	World Wide Web

ÚVOD

V dnešní době se čím dál tím více využívá internet věcí, který lidem v mnoha ohledech dokáže usnadňovat život. Jeden z prvních zástupců internetu věcí – předchůdce současných chytrých zařízení, automat na sodovku, vznikl roku 1982. Od ostatních tehdejších automatů se lišil tím, že byl připojen do ARPANETu (což byl předchůdce internetu) na dálku bylo možné zjistit obsah automatu a zda jsou nápoje vychlazeny na správnou teplotu. Vylepšení vytvořil jeden ze studentů se svými spolužáky proto, aby nemusel nadarmo podstupovat zdlouhavou cestu k onomu automatu. V té době ještě nikdo netušil, že takových zařízení tu do pár let bude mnohonásobně více, ke konci roku 2019 přibližně 14,2 miliard.

Díky tomu, že využívané komponenty jsou dnes technologicky rozmanitější, mnohem menších rozměrů a zároveň dostupnější než kdy dříve, proniká internet věcí téměř do všech oblastí lidského života. Využívá se v průmyslu, zemědělství, energetice, zdravotnictví, chytrých domácnostech, válečných konfliktech atd. Rapidně tak roste počet chytrých zařízení a s tím i nároky na jejich správu.

Cílem této bakalářské práce je vytvoření aplikace, která bude primárně sloužit jako podpůrný nástroj pro usnadnění práce se zařízeními Raspberry Pi používaných pro vzdálené měření a odečet energií. Teoreticky však může pracovat s jakýmkoliv zařízením podporujícím protokol SSH. Aplikace bude uživateli zjednodušovat vykonávání příkazů nad zařízeními, distribuci a správu softwaru pro jednotlivá zařízení a zajišťovat generování a správu certifikátů, které zařízení využívají k připojení do interní virtuální privátní sítě. Nyní jsou všechny tyto úkony vykonávány manuálně a jsou časově náročné. Hlavní benefit nového softwarového nástroje spočívá v tom, že dokáže formou paralelního procesování a šablon vykonávat procesy nad libovolným počtem zařízení zároveň. Cílem této práce je tedy vytvořit podpůrnou aplikaci, která primárně zjednoduší práci se zařízením a tím zkrátí čas na jeho správu.

V teoretické části práce bude popsáno, co je internet věcí, jeho historie a jak se využívá nyní. Čtenář bude také seznámen s virtuální privátní sítí a jejím využitím k anonymnímu a bezpečnějšímu pohybu na internetu. Budou zde také popsány technologie využité v praktické části práce. Platforma JavaFX využitá pro tvorbu uživatelského rozhraní, MySQL používané k uchovávání dat, Hibernate pro zajištění jednodušší práce s databází, knihovny Jsch nabízející usnadnění práce s SSH, a další. Cílem teoretické části práce je přiblížit čtenáři téma „Internet věcí“ a seznámit ho s technologiemi, které budou v praktické části práce využity.

V praktické části práce, se čtenář podrobněji seznámí se samotnými funkcemi a způsoby používání nástroje. Popisy budou doplněny screenshoty a ukázkami jakým způsobem funkce

správně využívat. Cílem praktické části práce je seznámit čtenáře se samotným nástrojem a ukázat mu, jak s ním lze samostatně pracovat.

1 INTERNET VĚCÍ

Internet věcí je v podstatě síť, která sdružuje chytrá mechanická, nebo digitální zařízení. Převážně to jsou vzájemně propojené jednoúčelové stroje. Každé z těchto zařízení má svůj jedinečný identifikátor, díky kterému ho lze odlišit od ostatních. Je také schopno samostatně komunikovat s ostatními zařízeními v síti, vyměňovat si s nimi data a dále s nimi kooperovat. To vše dokáže samo, bez lidského zásahu. Zjednodušeně řečeno to jsou zařízení schopna samostatně zpracovávat, odesílat a reagovat na informace z vnějšího světa.

„Internet věcí", z anglického „Internet of Things", zkráceně IoT, je v podstatě zastřešující pojem, který zahrnuje spoustu koncepcí a technologií. Paradoxem je, že základem internetu věcí nejsou věci, ale data, která tyto věci poskytují. Jedna z možností, jak termín chápat je, že se jedná o shromažďování, analyzování a následné reagování na informace, se kterými zmíněná chytrá zařízení pracují.[1]

Díky miniaturizaci a stále levnějším komponentám zažívá IoT v posledních letech obrovský rozmach. Je to také díky tomu, že právě shromažďovaní a analýza ohromného množství dat generovaných těmito zařízeními, přináší příležitost zhodnotit a zlepšit způsoby, jakým jsou poskytovány služby a produkovány produkty. [3]

1.1 Historie

Pojem Internet věcí se do širšího povědomí dostal v roce 1999.[2] Myšlenka propojených, spolupracujících zařízení je však mnohem starší. Již v roce 1926 řekl známý vědec a vynálezce Nikola Tesla toto:

„Když se bezdrátové připojení dokonale aplikuje, celá planeta se přemění v „obrovský mozek", což ve skutečnosti bude znamenat, že všechny věci budou částice reálného a rytmického celku. Nástroje, skrze které budeme toto schopni uskutečnit budou v porovnání s naším současným telefonem úžasně jednoduché. Člověk je bude bez problému schopný nosit v kapse jeho vesty." [2]

Samotná myšlenka tedy vznikla už 10 let předtím, než se začalo pracovat na prvním z počítačů s názvem Z1. V průběhu 20. Století, zejména pak se vznikem internetu v 70.-80. Letech, vystupovalo více odborníků s podobnými názory, vizemi a předpověďmi globálně propojeného světa, právě pomocí nesčetného množství zařízení, která zcela změní dosavadní způsob života.[2]

Za prvního zástupce „chytrých“ zařízení, je považován automat na sodovku, který v roce 1982 upravili studenti Univerzity Carnegieho–Mellonových. Stalo se tak na popud jednoho ze studentů, který byl otrávený z toho, že musí podstupovat zdlouhavou cestu k automatu s tím, že ho možná nalezne prázdný. Automat připojili do ARPANETu a díky tomu bylo možné na dálku zjistit jeho obsah a zda jsou lahve vychlazené na správnou teplotu. V té době nikdo z nich netušil, že vytvořili prvního zástupce „chytrých“ zařízení a že za pár desítek let jich budou ve světě miliardy. [5]

Do konce 20. století se objevilo více chytrých zařízení. Z těch nejznámějších například toaster, který bylo možné na dálku řídit přes síť, nebo kávovar, který byl monitorován a jeho obsah sdílen na firemní server. Nicméně šlo jen o ojedinělé případy, považované spíše za kuriozity. [2]

Do širšího povědomí termín IoT pronikl teprve v roce 1999, kdy díky propojení s novou technologie RFID a rapidnímu rozšiřování WWW přilákal více zájemců a možností využití. Za opravdový milník lze považovat roky 2008-2009, kdy celkový počet IoT zařízení překročil počet lidí na planetě Zemi. Ke konci roku 2019 existovalo přibližně 14,2 miliard zařízení a dle předpovědí odborníků bude tento počet dále růst. [2]

1.2 Chytrá zařízení

Za chytrá zařízení jsou považována elektronická zařízení, která do jisté míry chápou kontext informací, se kterými pracují. S těmito daty za pomocí autonomního procesování dokáží pracovat a mají schopnost komunikovat s ostatními zařízeními kabelově, nebo bezdrátově. [1]

Z předchozího textu vyplývá, že aby bylo možné zařízení definovat jako chytré, musí mít tři vlastnosti: vědomí svého vlastního kontextu, autonomní procesování a síťovou konektivitu. Jakákoliv věc se tedy může stát součástí IoT. Například pokud je k obyčejnému odpadkovému koši přidáno čidlo kontrolující úroveň plnosti, zařízení s menším výpočetním výkonem a síťovým připojením, lze takovýto koš považovat za chytré zařízení. [4]

1.2.1 Vědomí kontextu

Vědomí kontextu je schopnost systému shromažďovat informace o svém prostředí, ty dále vyhodnocovat a na jejich základě upravovat své chování. Za takový kontext může být považována jakákoli pro proces, který zařízení vykonává, relevantní informace, například čas, poloha, aktivita, jiná zařízení v blízkosti aj. [3]

Příkladem může být mobilní zařízení, které z uložených historických dat a jeho aktuálních GPS souřadnic dokáže zjistit, že se nachází poblíž restauračního podniku, který jeho uživatel v minulosti často navštěvoval. Pomocí notifikace ho informuje například o tom, že zde zrovna probíhá nějaká událost.

1.2.2 Autonomní procesování

Hlavním prvkem autonomního procesování je to, že zařízení dokáže zcela samostatně vykonávat, nebo řídit další zařízení tak, aby vykonávala nějakou funkci bez přímé interakce uživatele, či jiného zařízení.[3]

Za příklad lze považovat například klimatizaci, která je na základě informací o aktuální teplotě schopna tuto teplotu regulovat. Příkladem autonomního procesování je i notifikace na displeji mobilního telefonu, zmíněná v předchozí podkapitole, která se také zobrazí bez jakékoli přímé interakce s uživatelem.

1.2.3 Konektivita

Konektivita je schopnost zařízení připojovat se do datových sítí a sdílet nebo přijímat data od jiných zařízení. Tato schopnost je v IoT velmi důležitá a umožňuje sdružovat více menších zařízení do komplexnějších celků. Ty díky kooperaci dokážou být více efektivní. [3]

Příkladem jsou například chytré domácnosti. Díky informaci o čase s dostatečným předstihem vyhřejí místnosti tak, aby se majitel probouzel za příjemné pokojové teploty a zároveň neprotopil celou noc, zapnul se kávovar tak, aby měl kávu připravenou rovnou k pití, vytáhly žaluzie, zapnulo osvětlení, nastartovalo auto, aby se nahřál motor apod. Možností je nespočet a v dnešní době už se představivosti téměř meze nekladou.

1.3 Oblasti využití

V IoT existují dva hlavní směry: spotřebitelský a průmyslový. Hlavním zaměřením spotřebitelského internetu věcí jsou spotřebitelé a zvyšování jejich uživatelského zážitku. Průmyslový internet věcí se zaměřuje zejména na zefektivnění zařízení a systémů využívaných v průmyslových odvětvích. [1]

Další segment, který se v současné době rychle rozvijí, je bojový internet věcí. Jeho rozvoj je z velké části poháněn myšlenkou, že díky zapojení IoT, umělé inteligence a strojového učení, lze významně snížit čas mezi sběrem, analýzou a reakcí na informace proudící z bojiště.[6]

1.3.1 Chytré domy

Koncept chytrých domácností je o připojení co největšího počtu zařízení nacházejících se v domácnosti do internetu. Jedná se o zařízení od domácích spotřebičů a osvětlení přes dveřní zámky až po termostaty. Majitel je pak schopen monitorovat jejich stav a ovládat je vzdáleně. Získá také mnohem lepší přehled o stavu domácnosti, sníží náklady na provoz, zvýší se jeho komfort a celkově se zjednoduší život v takovémto domě. V případě jakéhokoli bezpečnostního incidentu, nebo havárie je majitel (případně bezpečnostní složky) informován a může rychleji reagovat předtím, než se vyskytnou větší škody na majetku. [3]

Existují různé realizace těchto systémů. Ve většině případu jsou jednotlivá zařízení připojená k centrální řídící jednotce. Ta zařízení spravuje, pracuje s daty, která poskytují a zajišťuje komunikaci s uživatelem. Uživatel pak se systémem může komunikovat například pomocí v domě rozmístěných terminálů, mobilní aplikace, nebo hlasovými pokyny. Uživatel se také, díky tomu, že systém bývá často připojen do internetu, může připojit ať už z domova, nebo z druhého konce světa. Nicméně komunikace se samotným uživatelem je spíše doplňková, prioritou je zautomatizování domácnosti a šetření zdrojů.[3]

1.3.2 Průmysl 4.0

Pod pojmem Průmysl 4.0 se skrývá jeden klíčový prvek – propojování. Ruku v ruce s ním jde i stále větší automatizace a optimalizace procesů v oblasti výroby, logistiky a služeb. První tři vývojové stupně průmyslové výroby se vyznačovaly inovacemi v oblasti mechaniky (Průmysl 1.0), elektroniky (Průmysl 2.0) a informačních technologií (Průmysl 3.0). Pro Průmysl 4.0 je typické, že se klasické průmyslové procesy stále více propojují se sdělovací a datovou technikou a vytvářejí automatizované, robotické výrobní systémy. Tím umožňují realizovat vizi samořízené výroby.[7]

Hlavní motivací k vytvoření iniciativy zvané Průmysl 4.0 je z pohledu výrobních společností potřeba zrychlit procesy, které je nutné vykonat k úspěšnému uvedení nových produktů na trh a současně rozšířit možnosti modifikací těchto produktů, dle požadavků jednotlivých zákazníků. To vše bez ztráty kvality a zároveň s větší úsporou zdrojů firmy.[8]

Se zaváděním těchto změn se nepromění jen průmysl samotný, ale dotkne se to mnoho dalších odvětví. Na společensko-sociální úrovni dojde k velkým proměnám, ať už z důvodu proměny pracovního trhu, tak i díky sběru a rychlému vyhodnocování personálních dat. Zlepší se tak cílení služeb, designu a vlastností vyráběných produktů dle potřeb jednotlivých skupin obyvatelstva i jednotlivců.[8]

1.4 Bezpečnost

Internet věcí má kromě spousty výhod i svou stinnou stránku a tou je bezpečnost. Vzhledem ke snahám výrobců, co nejrychleji a zároveň s co nejnižší pořizovanou cenou zpřístupnit chytrá zařízení zákazníkům, jde často bezpečnost stranou. [9]

Pokud zařízení bezpečnostními mechanismy disponují, často je jejich uživatelé nevyužívají z důvodu neznalosti, nebo jejich důležitost bagatelizují. Důsledky mohou být od ohrožení majetku jednotlivých osob, až po ochromení celých podniků.[9]

1.4.1 Rizika v domácí a osobní sféře

Narušení bezpečnosti v této oblasti může vést k ohrožení osobního majetku, a dokonce i přímo lidského zdraví.

K chytrým zařízením se řadí také zdravotnické pomůcky, například kardiostimulátory, insulinové pumpy apod. Pokud by někdo narušil, nebo upravil jejich funkčnost, může tím dojít závažným způsobem k ovlivnění zdraví jejich uživatele. Bezpečnostním rizikem mohou být i chytré automobily, a to tím, že někdo kompletně převezme kontrolu nad řízením vozidla.[10]

K ohrožení majetku může dojít deaktivací elektronických zámků, nebo bezpečnostního alarmu. Útočník může napadnout i další chytrá domácí zařízení, například lednici, či topení. Zablokuje je a bude požadovat úplatu za to, aby je znovu zprovoznil. Čím více uživatelé digitalizují svou domácnost, tím více by měli dbát na kybernetickou bezpečnost.[11]

1.4.2 Rizika v podnikové sféře

Donedávna bývaly komunikační datové toky průmyslových zařízení zcela odděleny od internetu, nebo probíhaly na specifických protokolech. Avšak v současné době vedly ekonomické přínosy, které souvisí hlavně s jednodušším využíváním obrovského množství průmyslových dat k tomu, že jsou tato zařízení připojována do sítě, využívající běžné internetové komunikační protokoly.[10]

V takovýchto případech pak platí, že bezpečnost sítě je na takové úrovni, jakou má nejhůře zabezpečený prvek. V jednom z útoků, při kterém útočníci získali neoprávněný přístup k datovému uložišti kasina a odcizili přes 10 gigabytu dat, využili akvária připojeného do interní sítě. [12] Spousta podobných zařízení je z pohledu bezpečnosti opomíjená, a proto právě díky nim vzniká nejvíce bezpečnostních incidentů, které dokážou ochromit chod celé firmy.

1.4.3 Iot botnety

Napadená zařízení mohou být využita i k jiným účelům, než je poškození svého majitele. Právě kvůli nižšímu zabezpečení jsou chytrá zařízení často zneužívána. Útočník zařízení infikuje škodlivým softwarem, díky kterému nad ním může převzít kontrolu. Infikované zařízení je připojeno k ostatním napadeným zařízením, povětšinou řízených z jednoho centra. Takto napadená skupina zařízení se nazývá botnet.

Výpočetní výkon zařízení zapojených do botnetu je pak útočníkem zneužíván. Převážně jde o nelegální činnost vedoucí k jeho finančnímu obohacení. Typické zneužití botnetů slouží k rozesílání SPAMu, provádění DdoS útoků s cílem odstavit určitou službu, či k dnes velmi populární těžbě kryptoměn.[13]

2 VIRTUÁLNÍ PRIVÁTNÍ SÍŤ

Základní myšlenkou VPN je vytvoření zabezpečeného šifrovaného připojení. Lze ho vytvořit mezi dvěma sítěmi, nebo mezi konkrétním uživatelem a sítí. Jedna z hlavních výhod této služby je zabezpečená komunikace mezi dvěma uzly přes nedůvěryhodné, potencionálně nebezpečné počítačové sítě. Zjednodušeně funguje tak, že zašifruje uživatelem přenášená data a ukryje ho za falešnou IP adresu. Umožňuje mu tak používat internet v soukromí.

Dalším způsobem využití je, že po připojení uživatele propojí se sítí dané organizace (intranetem). Ta mu umožní nerušeně využívat veškeré interní aplikace, či přístupy k jednotlivým soukromým službám, které za normálních okolností nejsou z veřejného internetu přístupné.[26]

2.1 Princip fungovaní

Společnost, či osoba poskytující službu VPN musí mít řádně nastaven a spuštěn VPN server, na druhé straně musí mít uživatel nainstalovaného VPN klienta. Nejprve je nutné, aby se uživatel autentizoval. Důvěryhodnosti během autentizačního procesu se nejčastěji dosahuje použitím digitálního podpisu, nebo šifrováním ověřovací zprávy předem sdíleným klíčem.

Po ověření identity klienta se mezi ním a serverem vytvoří tunel. Data jsou na straně klienta nejprve zašifrována VPN klientem a poslána tunelem na vybraný VPN server. VPN server data rozšifruje a přepošle data cílovému serveru. Totéž platí i pro komunikaci opačným směrem. Případný útočník tedy neví, s kým ve skutečnosti uživatel komunikuje, a ani cílový server neví z jaké IP adresy přichází komunikace. Obě strany, útočník a cílový server, vidí pouze adresu VPN serveru.

Nicméně data, přestože jsou zašifrovaná, nejsou v žádném případě v nedůvěryhodné části sítě dokonale chráněna. Důvěra mezi VPN klientem a VPN serverem je zde naprosto zásadní faktor, který je nutné určitým způsobem vnímat jako potencionální bezpečnostní hrozbu. Poskytovatel VPN služby může také používat vzdálené servery, nad kterými nemá fyzickou kontrolu, což lze považovat za určitou mezeru v celkové bezpečnosti.

Riziko může nastat i v samotné sítí skryté za VPN serverem. Příkladem může být použití protokolu http, kde jsou data po dobu přenosu z VPN serveru na cílový server čitelná, protože šifrování je zprostředkováno pouze v tunelu mezi klientem a VPN serverem. Využití protokolu HTTPS je zde perfektním doplněním k technologii VPN, kde jsou veškerá koncová data šifrována po celou dobu přenosu.[26]

2.2 OpenVPN

Jedná se o open-source VPN protokol. To, že je dostupný i v podobě zdrojového kódu, je jeho hlavní výhodou a zároveň i nevýhodou. Každý může přispět a vylepšit jeho podobu, ale zároveň může prozkoumávat jeho slabiny.

I přesto je ale OpenVPN jeden z nejpoužívanějších a zároveň nejbezpečnějších VPN protokolů. Podporuje vysoké úrovně šifrování, jako například 256bitové šifrování, které je v současné době považované za nejbezpečnější, 2048bitové RSA asymetrické šifrování a další šifrovací metody. Ovšem i zde platí, že čím lepší zabezpečení, tím pomalejší je ve výsledku přenos dat.

Výhodou je použitelnost na všech předních platformách, a dokonce i na některých méně podporovaných jako je například Raspberry Pi. K dispozici je také velké množství návodů, včetně modifikací pro různé použití. Velkým plusem je, že jeho použití je zcela zdarma.[26]

2.3 Využití

Jednou z velmi ceněných vlastností VPN je ochrana soukromí. Otevírá možnost bez obav využívat veřejné Wi-Fi sítě a to tím, že hackeři mají snížené šance na narušení soukromí. Z bezpečnostního hlediska je téměř nemožné považovat některý způsob komunikace za maximálně bezpečný. Přesto lze tento způsob komunikace označit za velmi dobře zabezpečený a lze díky němu vystupovat na veřejném internetu anonymně.

Další výhodou je, že není zapotřebí speciálního hardwaru, což snižuje cenové nároky na pořízení této služby. Samotné připojení k VPN službě není zdlouhavé, ani náročné a zvládne jej i technicky méně zdatný uživatel.

Jednou z dalších poměrně často využívaných výhod je obejití lokální cenzury. Stává se, že v některých státech, je poskytovatelům internetu vládou nařízeno, aby blokovali vstup na některé internetové služby. S pomocí VPN lze tento problém vyřešit.[24]

2.4 Nevýhody

I přes veškeré uvedené výhody, je nutné připomenout i některá úskalí. Jedním z nich je například zhoršení rychlosti přenosu dat, a to o cca 5-10% původní rychlosti. Záleží na zvoleném tarifu u poskytovatele VPN služby a kapacitách jeho datových center.

Pokud je pro uživatele důležité mít internetové připojení s co nejnižší odezvou, je připojení VPN nevhodné. I přesto, že se může jednat pouze o desítky milisekund, například v případě

online her jde o velmi zásadní hodnoty, které mohou herní zážitek velmi znepříjemnit. Z toho důvodu je doporučeno v době hraní VPN nepoužívat.

Doporučuje se také nepoužívat VPN, které jsou poskytované zdarma. Provoz serverů, které dokážou zamaskovat datový provoz uživatele samozřejmě něco stojí. Je tedy vhodné za lepší ochranu soukromí zaplatit. Nejedná se navíc o příliš vysoké částky. [25]

3 POUŽITÉ TECHNOLOGIE

V implementaci bakalářské práce je použito několik technologii a frameworků. V této kapitole budou krátce představeny nejdůležitější z nich a bude uvedeno jejich začlenění v samotné práci.

3.1 Java

Java je univerzální vysoko-úrovňový programovací jazyk, který je založen na třídách, objektově orientovaný a navržený tak, aby měl co nejméně závislostí. Účelem je umožnit vývojářům aplikaci „napsat jednou, spouštět kdekoli“ (WORA), což znamená, že kompilovaný kód Java může běžet na všech platformách, které podporují Javu, aniž by byla nutná rekompilace. Java aplikace jsou obvykle kompilovány do bajtkódu, který může běžet na jakémkoli virtuálním stroji Java (JVM) bez ohledu na základní architekturu počítače.[14]

V práci je využit jako primární programovací jazyk.

3.2 JavaFx

JavaFX je framework pro tvorbu vizuálně bohatých okenních aplikací. Ve frameworku je kladen důraz na jednoduchost tvorby. JavaFX se hodí jak pro desktopové aplikace, tak pro webové applety nebo mobilní aplikace. Má podporu v systémech Microsoft Windows, Linux, Android a MacOS.[15]

V práci je využíván k tvorbě GUI ve formě architektonického vzoru MVC.

3.2.1 Gluon Scene Builder

Scene Builder je nástroj pro vizuální rozvržení, který umožňuje uživatelům bez kódování rychle navrhovat uživatelská rozhraní aplikací JavaFX.

Uživatelé mohou přetahovat komponenty pomoci metody drag-and-drop, upravovat jejich vlastnosti a používat šablony stylů. Samotný kód FXML se automaticky generuje na pozadí.

Výsledkem je soubor FXML, který lze poté zkombinovat s Java projektem vazbou uživatelského rozhraní na logiku aplikace. [16]

3.3 Hibernate

Hibernate je framework v jazyce Java, umožňující objektově-relační mapování (ORM). Implementuje Java Persistence API. Může být použito buď specificky – tak, že jsou volány přímo jeho metody, nebo obecně přes rozhraní definované JPA.

Hibernate dále vyvíjí speciální jazyk pro psaní dotazů do databáze přímo z kódu, tzv. Hibernate Query Language (HQL). HQL je inspirován syntaxí SQL, takže je dobře použitelný pro každého, kdo zná základy SQL dotazů. [17]

V práci je použit jako hlavní podpůrný framework pro práci s databází.

3.3.1 Objektově relační mapování

Objektově relační mapování neboli ORM, je programovací technika, která umožňuje ukládat data objektově orientovaných jazyků do relační databáze. Vybraným třídám pak přísluší databázové tabulky, přičemž datové typy se konvertují na SQL datové typy. ORM zpracovává také vztahy mezi objekty, které do relačních databází uživatel ukládá. V databázi jsou pak vztahy objektů reprezentovány spojováním tabulek cizími klíči. [17]

3.4 MySQL

MySQL je multiplatformní open-source databáze. Využívá systém pro správu relačních databází (RDBMS) s modelem klient-server. RDBMS je software nebo služba používaná k vytváření a správě databází na základě relačního modelu. Pro svou snadnou implementovatelnost, výkon, a především díky tomu, že se jedná o volně šiřitelný software, je v současné době hodně využíván. [18]

V práci je použita jako primární uložiště dat.

3.5 Jsch

JSch je zkratka pro Java Secure Shell. Je to knihovna, která umožňuje připojení se ke vzdálenému serveru SSH. Poskytuje podporu pro bezpečné vzdálené přihlášení, bezpečný přenos souborů atd. Může automaticky šifrovat, autentizovat a komprimovat přenášená data. [19]

V práci zajišťuje nízko úrovňovou komunikaci přes protokol SSH.

3.5.1 SSH

SSH, také známý jako Secure Shell, je síťový protokol používající model klient-server, který uživatelům poskytuje bezpečný způsob přístupu k počítači přes nezabezpečenou síť. Kromě poskytování zabezpečených síťových služeb se SSH týká sady nástrojů, které implementují protokol SSH.

Secure Shell poskytuje autentizaci pomocí hesla, autentizaci pomocí veřejného klíče a také šifrovanou datovou komunikaci mezi dvěma počítači připojujícími se přes nezabezpečenou síť, jako je například internet. Kromě zajištění silného šifrování je SSH široce používán správci sítě pro vzdálenou správu systémů a aplikací, což jim umožňuje bezpečně se přihlásit k jinému počítači v síti, provádět příkazy a přesouvat soubory z jednoho počítače do druhého.[20]

3.6 ExpectIt

ExpectIt je knihovna v Javě, která usnadňuje a do jisté míry automatizuje interakci programu s jinými procesy, které využívají textového terminálu jakožto rozhraní pro komunikaci. Podporuje například práci s technologiemi Telnet, FTP, SFTP, SSH atd.

Objekt Expect vytvoří pro vstupní datový tok (InputStream) vlastní vlákno, které ze streamu načítá byty a ukládá je do String bufferu. Uživatel je pak na základě obsahu bufferu schopen reagovat na vstupní data. Hlavní výhodou je, že uživatel nemusí přímo pracovat s nízko-úrovňovou implementací jednotlivých technologií.[21]

V práci je využívána k práci s kódem vytvořeným uživatelem, hlavně pro práci s očekávanými výstupy.

3.7 Logback

Logování je klíčovou součástí jakékoli aplikace pro účely ladění a správy. Logback je framework zajišťující lepší práci s logováním v Java aplikacích, vytvořený jako nástupce populárního projektu Log4j. Je jedním z nejpoužívanějších logovacích frameworků v Javě. Nabízí rychlejší implementaci než Log4j, poskytuje více možností konfigurace a větší flexibilitu při archivaci starých logů.[22]

V práci je použit k logování do konzole a rotujícímu logování do souboru.

3.8 Maven

Maven je systém pro správu a sestavování aplikací postavených nad platformou Java. Jeho využitím odpadá závislost na konkrétním IDE, protože všechny informace potřebné ke kompilaci a sestavení programu jsou přímo obsaženy ve speciálních POM souborech. Systém Maven je již plně integrován do všech velkých IDE (Eclipse, Netbeans, IDEA) a tak je práce s ním opravdu velmi snadná i přes uživatelské rozhraní.

Principem systému Maven je vytvoření objektového modelu nad zdrojovým kódem, se kterým lze provádět různé operace. Nejčastěji jde o kompilaci, kontrolu a vytvoření aplikačních balíčků. [23]

V práci je využit hlavně ke správě závislostí a sestavování aplikačního balíčku.

4 NÁVRH A IMPLEMENTACE VLASTNÍ APLIKACE

4.1 Aktuální stav

Praktická část práce se zaměřuje primárně na práci se zařízeními Raspberry Pi, která jsou využívána pro monitoring energií. Přesněji řečeno, na těchto zařízeních běží software, který vykonává tuto činnost. Zařízení jsou připojena k internetu, který využívají ke sdílení informací. Zároveň jsou, za účelem lepšího zabezpečení komunikace, připojena do virtuální privátní sítě. K autentizaci zařízení v rámci virtuální privátní sítě jsou využívány certifikáty, vydávané vlastním VPN serverem.

V průběhu času se může vyskytnout potřeba upravit tento software, například za účelem změny funkčnosti, zlepšení bezpečnosti apod. Po změnách v monitorovacím softwaru je momentálně postupováno tak, že upravený software je na jednotlivá zařízení nahráván manuálně, bez využití jakékoliv automatizace.

Stejným způsobem je postupováno i v případě připojení zařízení do virtuální privátní sítě. Nejprve je nutné manuálně na VPN serveru sestavit samotný certifikát pro zařízení. To zahrnuje projití celého konfiguračního procesu nového certifikátu, při kterém jsou vygenerovány jeho části. Z vygenerovaných částí se pomocí textového editoru sestaví úplný certifikát. Na VPN serveru je ještě nutné nastavení statických IP adres pro zařízení. Následuje manuální nahrání vygenerovaného certifikátu na zařízení.

K zařízením i VPN serveru je přistupováno přes textový terminál pomocí SSH protokolu. Oba procesy jsou časově náročné, obzvlášť pak pokud je nutno operace vykonat pro větší množství zařízení. Proces, při kterém jsou manuálně sestavovány certifikáty, je náchylný na chyby, které pak většinou vedou k tomu, že je uživatel nucen vytvářet certifikát znovu.

4.2 Návrh aplikace

Program by měl sloužit jakožto podpůrný nástroj pro zařízení, podporující komunikaci prostřednictvím protokolu SSH. Primárně by měl zjednodušovat výměnu softwarových balíčků na zařízeních a správu zařízení připojených v rámci vlastní virtuální privátní sítě, která zahrnuje generování a distribuci certifikátů používaných k autentizaci zařízení.

Požadavkem je, že tyto procesy by měly být vykonávány ze samostatné aplikace. Uživatel tak nebude muset pracovat s textovým terminálem. Aplikace bude umožňovat ukládání informací o zařízeních a jednotlivých procesech do databáze. Také bude umožněno vykonávat samotné procesy nad více zařízeními zároveň, ať již sériovým, či paralelním procesováním.

Aplikace je vytvářena v platformě JavaFX jakožto okenní aplikace. Implementuje architekturu MVC, která rozděluje datovou část, uživatelské rozhraní a řídící logiku do tří nezávislých komponent se snahou o to, aby modifikace některé části měly jen minimální vliv na ostatní komponenty.

Aplikace se sestává z kontejneru karet. Je tedy dělená na jednotlivé karty obsluhující některou z funkcí.

Karta „Settings“ obsahuje konfiguraci databázového připojení, export/import dat do databáze, konfiguraci VPN serveru a informace potřebné pro tvorbu certifikátů a možnost volby druhu procesování.

Karta „Templates“ obsahuje seznam a možnost konfigurace procesů, ty se dále dělí na procesy „Firmware swap“ a „Custom code“.

Karta „Devices“ obsahuje seznam zařízení, možnost přidání a odebrání jednotlivých zařízení a spuštění procesu nad vybranými zařízeními. Z karty „Devices“ se lze dostat na detail zařízení, ze které je možné modifikovat informace o zařízení, kontrolovat jeho dostupnost, generovat a nahrát certifikát na zvolené zařízení. Na kartě „Devices“ lze vybrat zařízení a proces, který se nad nimi bude vykonávat. Tímto se uživatel dostane na kartu procesů, ve kterém je možné nad zařízeními proces spustit a sledovat jeho průběh.

4.2.1 Funkční požadavky

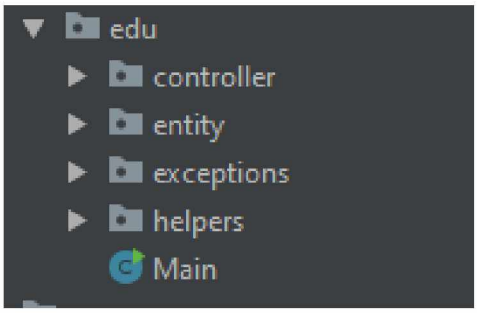
- Uchování a modifikace údajů o spravovaných zařízeních
- Uchování a modifikace údajů o procesech vykonávaných nad zařízeními
- Správa, generování a nahrání VPN certifikátů na zařízení
- Spouštění procesů na více zařízeních – sériově, nebo paralelně

4.2.2 Nefunkční požadavky

- Aplikace je spustitelná na systému Windows, nebo Linux.
- Program by neměl v paměti zabírat více než 500 MB
- Možnost propojení s lokální, nebo vzdálenou MySQL databází

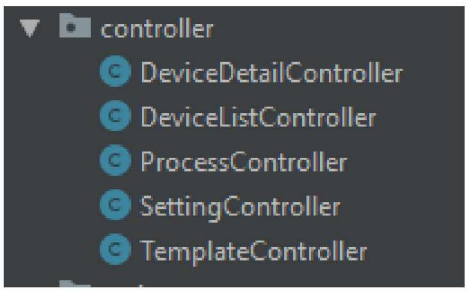
4.2.3 Struktura balíčků

Aplikace je vnitřně členěná na tyto balíčky:



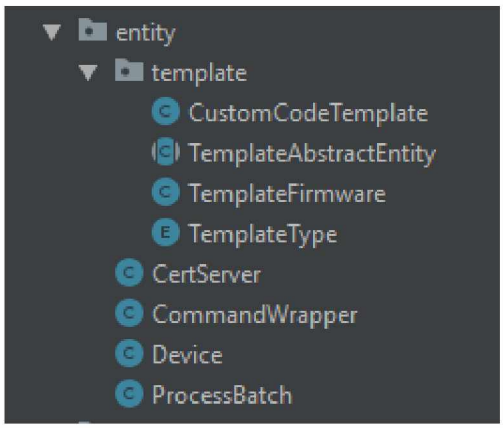
Obrázek 1: Balíček edu

- edu – hlavní balíček obsahující spouštěcí bod aplikace, třídu *Main*, a ostatní balíčky (Obr. 1)



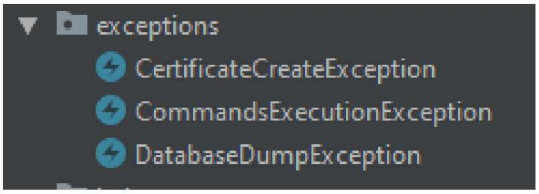
Obrázek 2: Balíček controller

- controller – balíček obsahující třídy, které slouží k obsluze dat a obsahují aplikační logiku (Obr. 2)



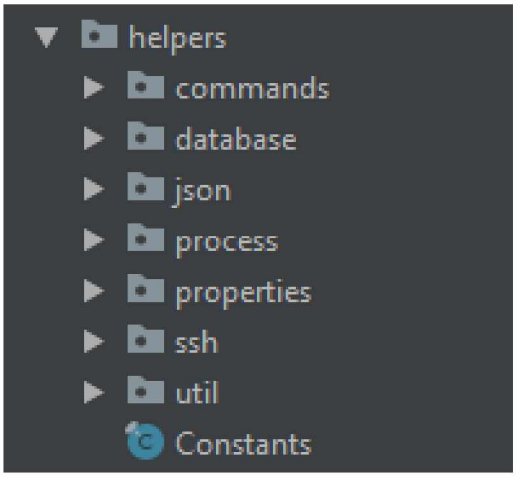
Obrázek 3: Balíček entity

- entity – balíček obsahující třídy, které reprezentují entity a obsahují informace o nich (Obr. 3)



Obrázek 4: *Balíček* exceptions

- exceptions – balíček obsahující uživatelsky definované výjimky (Obr. 4)



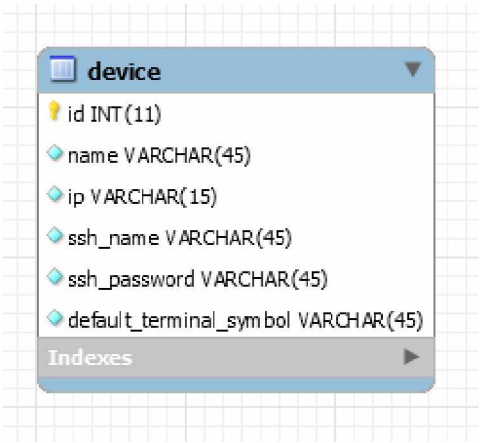
Obrázek 5: Balíček helpers

- helpers – balíček obsahující balíčky s podpůrnými třídami, pomocné například při práci s databází, SSH spojením, atd. (Obr. 5)

4.3 Návrh databázového schématu

K přechovávání dat je využita databáze MySQL, v databázi jsou uložena data šablon procesů a jednotlivých zařízení.

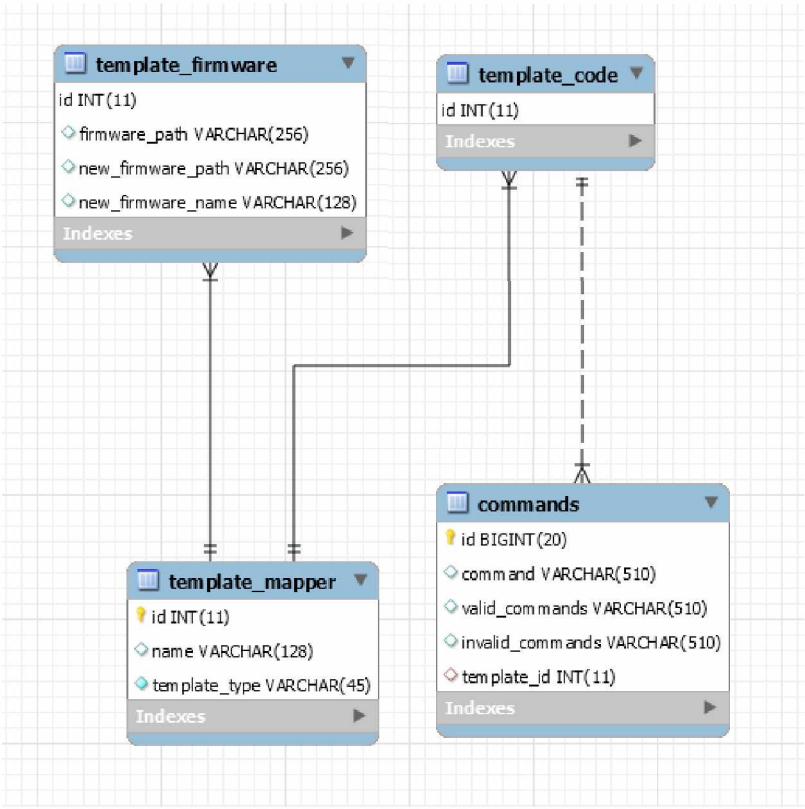
4.3.1 Tabulka zařízení



Obrázek 6: Tabulka device

Jednoduchá tabulka „device“ obsahuje informace sloužící k připojení a obsluze jednotlivých zařízeních spravovaných aplikací (Obr. 6).

4.3.2 Tabulky šablon



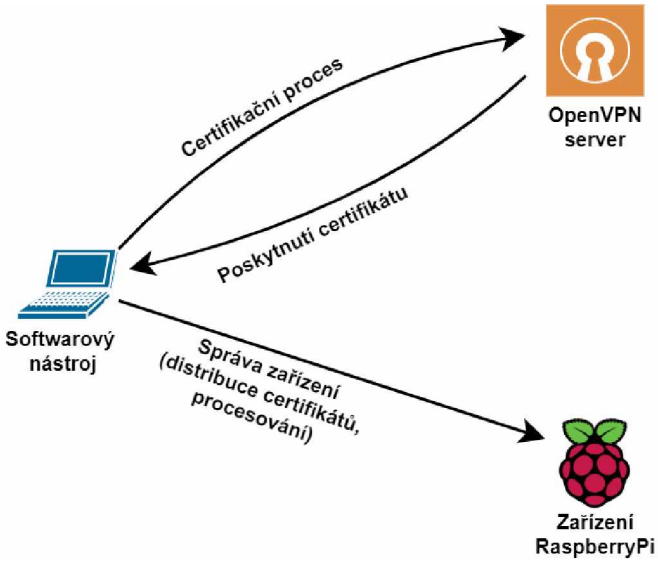
Obrázek 7: Tabulky šablon

V aplikaci existují dva druhy šablon. Jedna pracuje s výměnou softwarových balíčků, druhá umožňuje spustit libovolnou posloupnost příkazů. Implementovány jsou ve dvou třídách *FirmwareTemplate* a *CustomCodeTemplate*, které obě dědí z jedné abstraktní třídy *TemplateAbstractEntity*, tento vztah se promítá i do databázového schématu (Obr. 7).

Hlavní tabulkou je „template_mapper", která reprezentuje abstraktní třídu a obsahuje sdílené atributy pro jméno a druh šablony. Tabulky „template_firware" a „template_code" reprezentují jednotlivé podtřídy. S tabulkou „template_mapper" jsou ve vztahu 1:1 a jsou mapovány pomocí sloupců „id". Na tabulku „template_code" je navázána tabulka "commands" ve vztahu 1:m reprezentující jednotlivé příkazy obsažené v šabloně pro vykonávání libovolné posloupnosti příkazů.

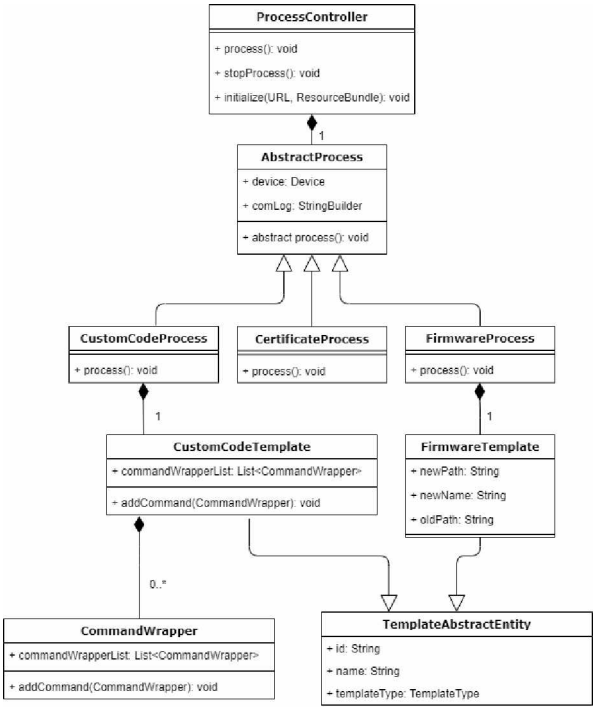
4.4 Distribuční část softwaru

V této kapitole bude čtenář blíže seznámen s primárním účelem aplikace, a to správou zařízení a vykonáváním procesů nad nimi.



Obrázek 8: Komunikační schéma procesování

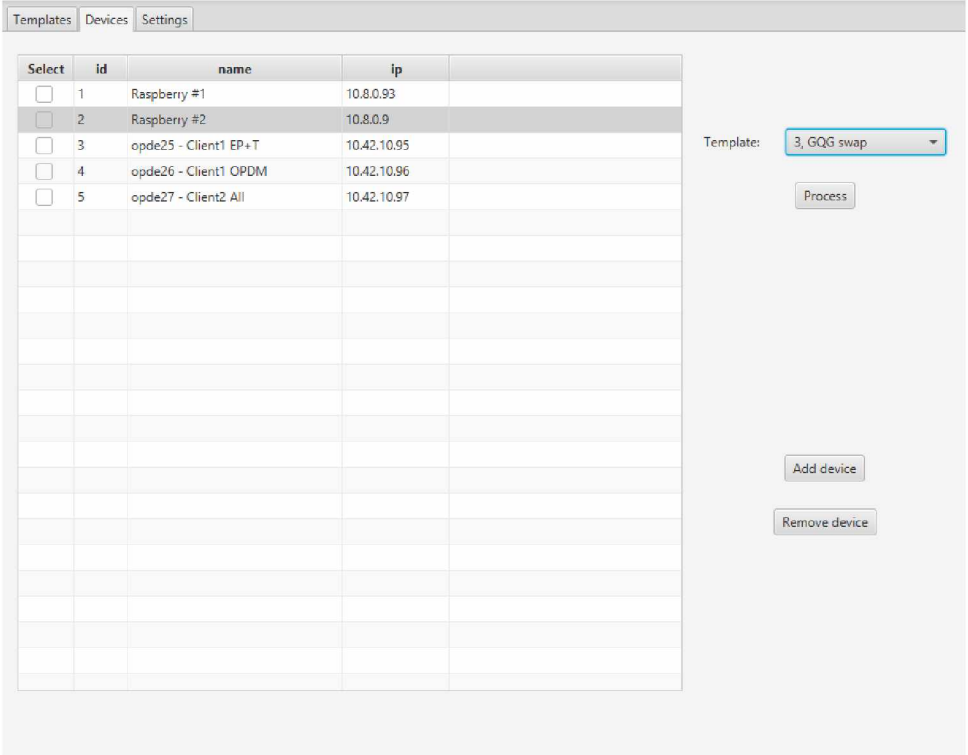
Obrázek 8 představuje schématickou reprezentaci prostředí, ve kterém softwarový nástroj pracuje a procesů, které jsou v něm vykonávány.



Obrázek 9: UML diagram distribuční části

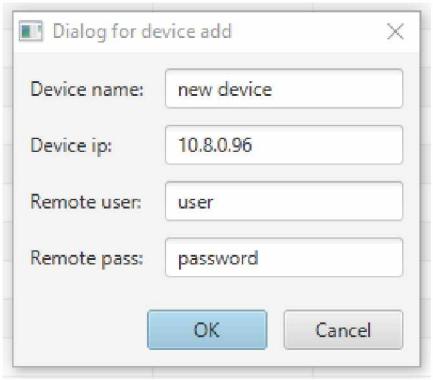
Na obrázku 9 lze vidět UML diagram tříd, použitých k realizaci distribuční části.

4.4.1 Správa zařízení



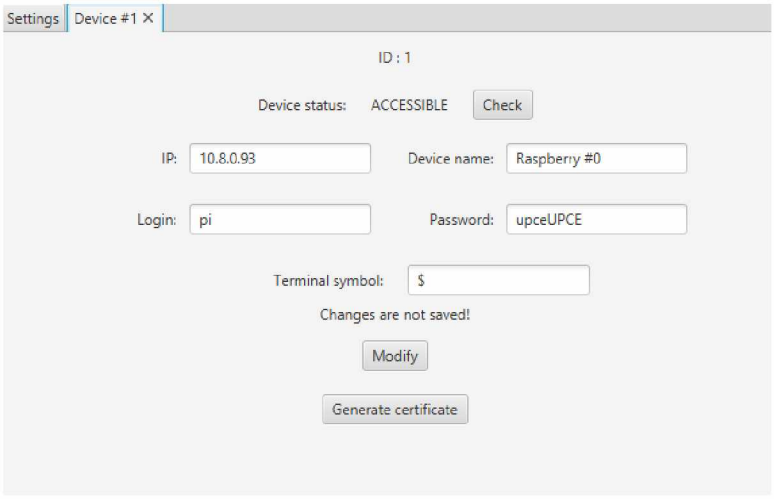
Obrázek 10: Karta „Devices“

Karta „Devices“ slouží primárně ke správě zařízení (Obr. 10). Převážnou část karty zabírá seznam se zařízeními, který zobrazuje nejdůležitější informace, podle kterých lze zařízení snadno identifikovat. Po pravé straně se pak nachází panel s obslužnými tlačítky.



Obrázek 11: Dialog pro přidání nového zařízení

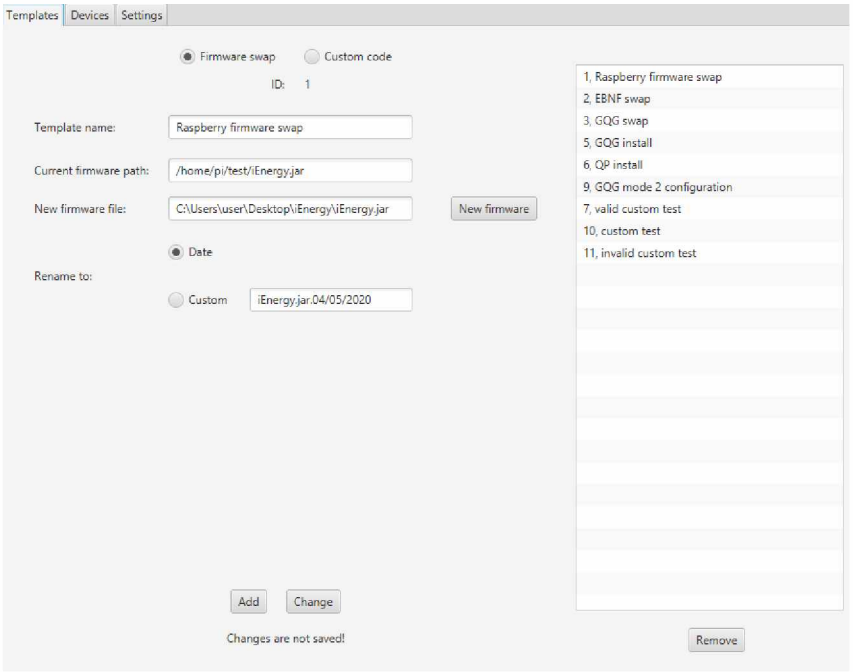
Tlačítko „Add device“ vyvolá dialogové okno (Obr. 11), kterým je možné přidat nové zařízení. Pomocí pravého tlačítka myši lze zvolit zařízení, ty pak mohou být odstraněna tlačítkem „Remove device“. Tlačítkem „Process“ je možné nad vybranými zařízeními provést proces vybraný z roletky šablon nacházející se nad tlačítkem.



Obrázek 12: Karta detail zařízení

Po dvojkliku myší na zařízení ze seznamu se otevře nová karta s detailem vybraného zařízení (Obr. 12). V této kartě lze upravovat informace o zařízení, kontrolovat dostupnost zařízení a generovat certifikáty pro VPN síť, využívající nastaveného VPN serveru. Při detekování změn v datech se zobrazí hláška, že změny nejsou uloženy, která po uložení opět zmizí. Pole s terminálním symbolem hraje důležitou roli v provádění procesů nad zařízením. Je důležité ho nastavit správně podle textového terminálu zařízení.

4.4.2 Správa šablon

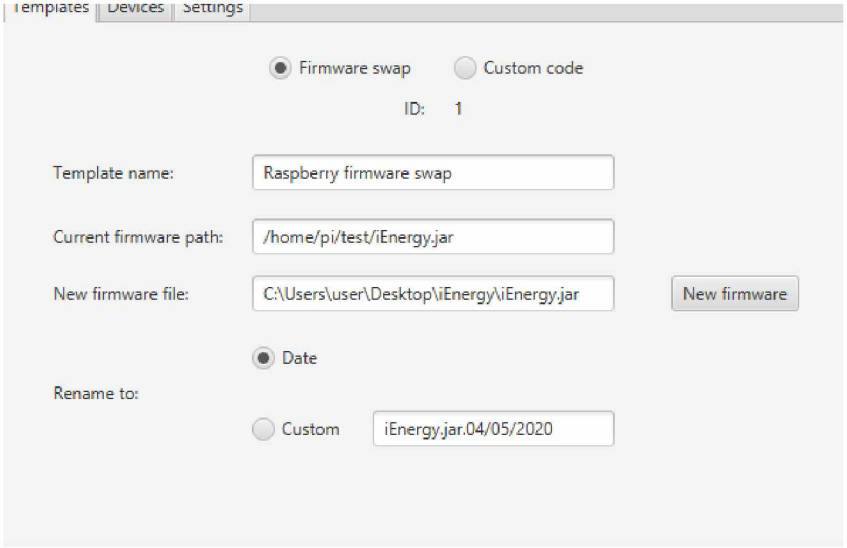


Obrázek 13: Karta „Templates“

Karta „Templates“ slouží ke správě šablon procesů vykonávaných nad zařízeními (Obr. 13). Po pravé straně se nachází seznam se šablonami. Šablony z toho seznamu jsou přítomny i v roletce v kartě „Devices“. Po kliknutí na šablonu v seznamu se načtou data o vybrané šabloně. Ve spodní části se pak nachází tlačítka pro přidání, upravení, nebo odstranění šablony. Při detekování změn v datech se zobrazí hláška, že změny nejsou uloženy, která po uložení opět zmizí.

V aplikaci existují dva druhy šablon – „Firmware swap“ a „Custom code“. „Firmware swap“ pracuje s výměnou softwarových balíčků, „Custom code“ umožňuje spustit libovolnou posloupnost příkazů. Lze mezi nimi přepínat pomocí elementu „RadioButton“ v horní části. Oba druhy šablon mají sdílené pouze „Template name“, konfigurace se pak pro každou z nich liší.

Šablona pro výměnu softwarových balíčků



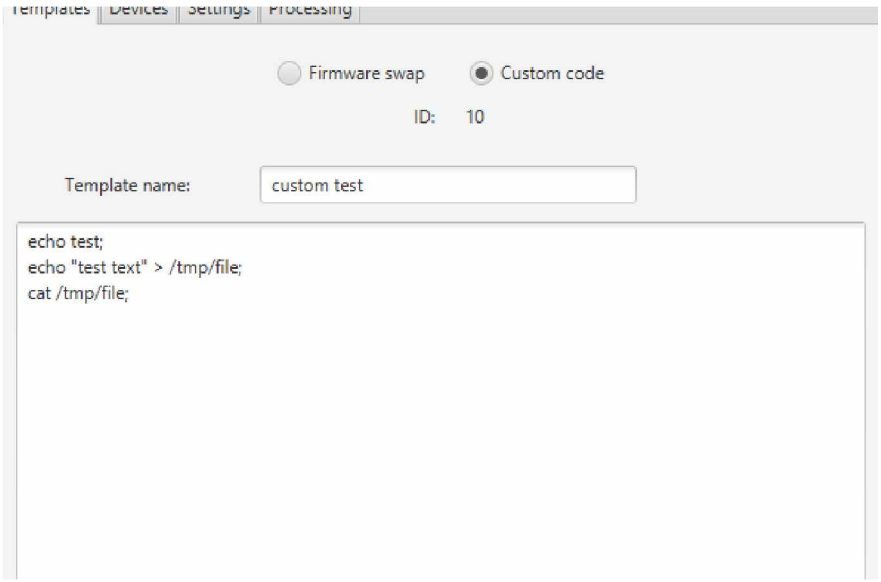
Obrázek 14: Konfigurace šablony „Firmware swap“

U tohoto druhu šablony (Obr. 14) je nutné vybrat cestu na vzdáleném zařízení k původnímu souboru, který bude nahrazen. Cesta je specifikována v „Current firmware path“.

Dále ze zařízení, kde je spuštěn nástroj, nastavit cestu k novému souboru, který bude nahrazovat původní soubor. Cesta je specifikována v „New firmware file“, lze ji pohodlněji načíst pomocí klasického dialogu pro načítání souborů pomocí tlačítka „New firmware“.

V poslední části je nutné vybrat, jakým způsobem bude přejmenován původní soubor. Lze vybrat mezi přejmenováním tím způsobem, že se za původní jméno souboru připojí datum, nebo přejmenováním na zcela nové, uživatelsky zvolené, jméno souboru.

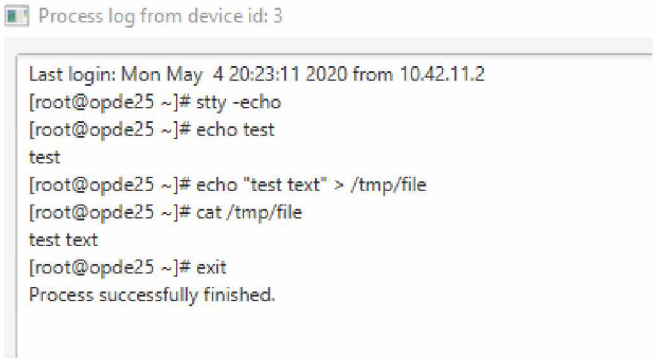
Šablona pro spuštění libovolné posloupnosti příkazů



Obrázek 15: Konfigurace šablony „Custom code“

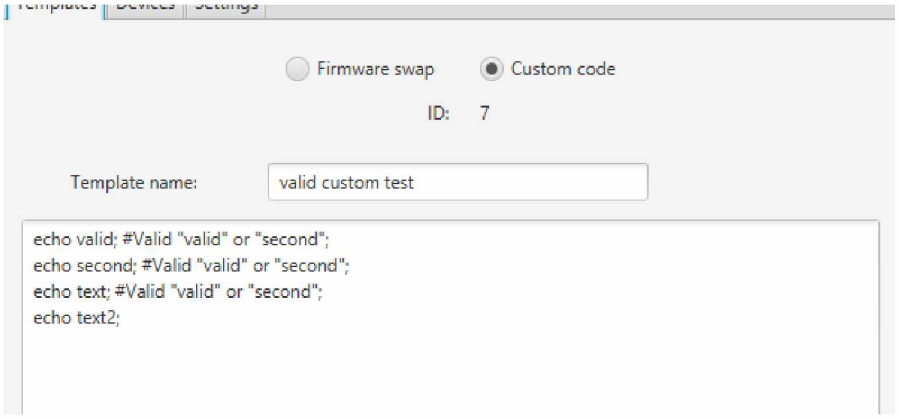
Pro konfiguraci tohoto druhu šablony (Obr. 15) je využito pouze jedno textové pole, do kterého mohou být zapsány standardní příkazy, které jsou spustitelné na cílovém zařízení. Jednotlivé příkazy musí být odděleny pomocí středníku.

V běžném režimu se mezi vykonáváním příkazů vždy čeká, než se na výstupu objeví terminální symbol, který je nastavený individuálně pro každé zařízení. Tímto se simuluje běžné vykonávání příkazů a je dosaženo toho, že jednotlivé příkazy jsou validně dokončeny a nejsou jednorázově všechny vloženy do terminálu.



Obrázek 16: Výstup procesu pro šablonu z obrázku č. 15

Chování šablony, respektive vykonávání příkazů, lze ovlivnit pomocí speciálních zápisů za jednotlivými příkazy. Zápis se skládá z klíčového slova „#Valid“, nebo „#Invalid“, očekávaných textových řetězců zapsaných ve dvojitých uvozovkách. V případě více očekávaných řetězců se oddělují klíčovým slovem „or“. Zápis se ukončuje středníkem.

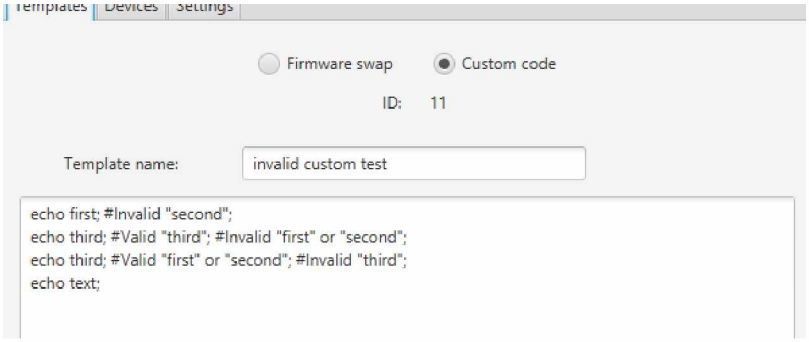


Obrázek 17: Konfigurace šablony „Custom code“ s využitím zápisu #Valid

Pomocí zápisu s klíčovým slovem „#Valid“ (Obr. 17) lze ovlivnit běh procesu tak, že vykonávání dalších příkazů bude podmíněno výstupem předcházejícího příkazu. Pokud výstup příkazu bude obsahovat jeden z očekávaných řetězců, proces bude pokračovat. Pokud se na výstupu ani jeden z řetězců neobjeví, po vypršení časového limitu bude proces předčasně ukončen.

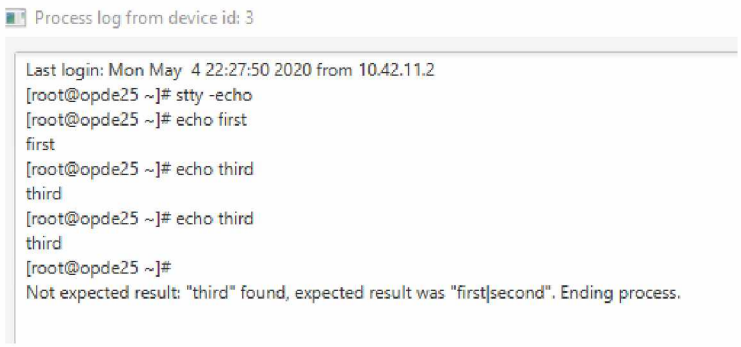
Obrázek 18: Výstup procesu pro šablonu z obrázku č. 17

Z výstupu (Obr. 18) je vidět, že poslední příkaz „echo test2“ nebyl na zařízení spuštěn a proces se ukončil předčasně.



Obrázek 19: Konfigurace šablony „Custom code“ s využitím zápisu #Invalid

Pomocí zápisu s klíčovým slovem „#Invalid“ (Obr. 19) lze ovlivnit běh procesu tak, že vykonávání dalších příkazů bude podmíněno výstupem přecházejícího příkazu. Pokud výstup příkazu nebude obsahovat ani jeden z očekávaných řetězců, proces bude pokračovat. Pokud se však na výstupu objeví nějaký z řetězců, proces bude ukončen.

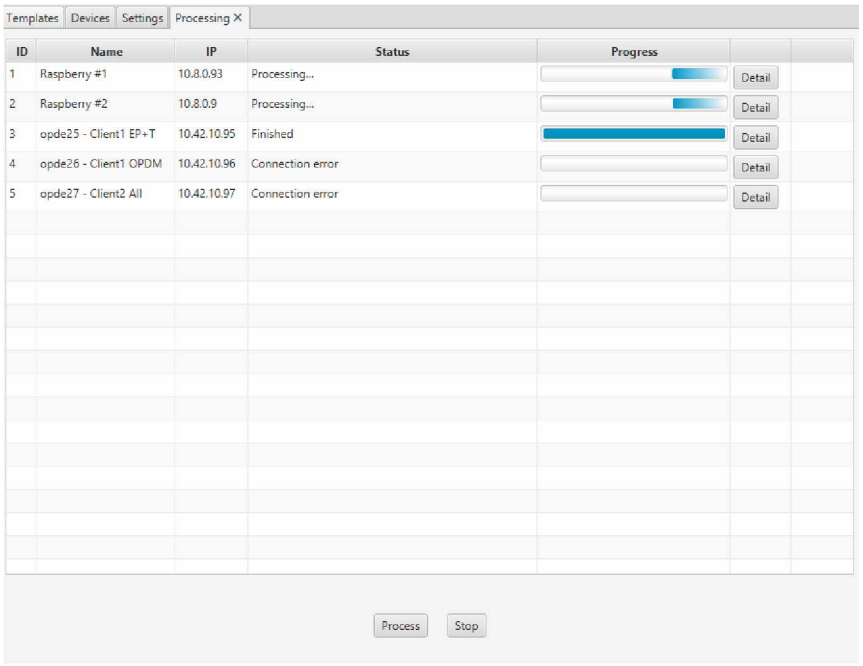


Obrázek 20: Výstup procesu pro šablonu z obrázku č. 17

Z výstupu (Obr. 20) je vidět, že poslední příkaz „echo test“ nebyl na zařízení spuštěn a proces se ukončil předčasně.

4.4.3 Procesování šablon

Po vybrání zařízení, volbě šablony procesů a kliknutí na tlačítko „Process“ v kartě „Devices“ se otevře nová karta „Processing“ (Obr. 21).

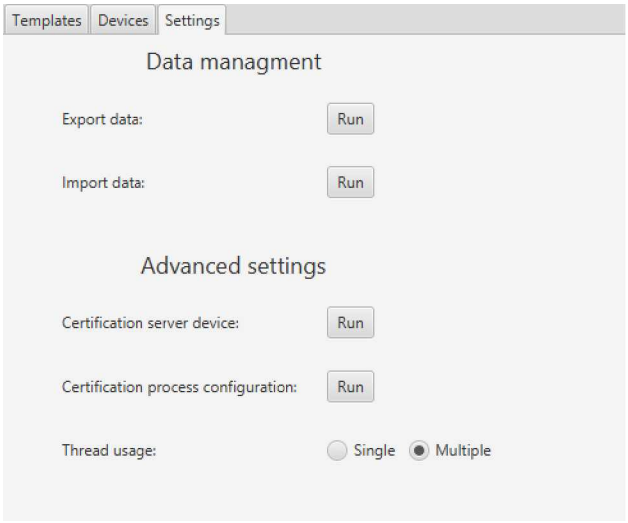


Obrázek 21: Karta „Processing“

Na této kartě jsou vidět informace o vybraných zařízeních. Po spuštění procesu lze pozorovat stav průběhu. Po kliknutí na tlačítko „Detail“ se pak otevře vyskakovací okno s detailním textovým záznamem průběhu procesu nad vybraným zařízením. Viz obrázky 14., 16. a 18.

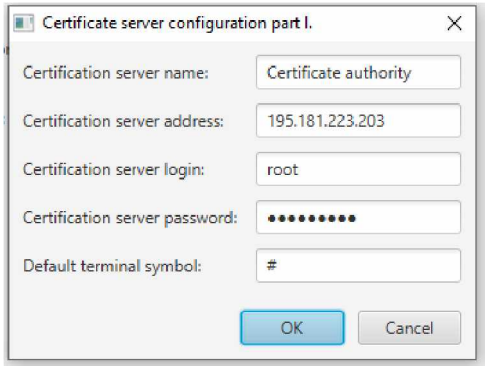
4.5 Správce VPN certifikátů

4.5.1 Konfigurační část



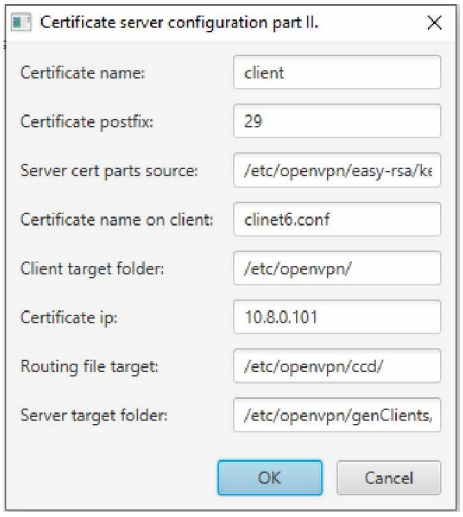
Obrázek 22: Karta „Settings“, detail na sekci „Advanced settings“

Předtím, než je možné v aplikaci generovat a nasazovat certifikáty, je nutné vyplnit informace o certifikačním VPN serveru a údaje nutné pro tvorbu certifikátu. Toto je možné nastavit v kartě „Settings“ v sekci „Advanced settings“ (Obr. 22).



Obrázek 23: Dialog pro nastavení VPN serveru

V první části (Obr. 23) se vyplňují informace o VPN serveru, které jsou v podstatě obsahově podobné, jako ty, co se vyplňují u jednotlivých zařízení.



Obrázek 24: Dialog pro nastavení VPN serveru

V další části (Obr. 24) se pak nastavují informace nutné ke správnému vytvoření certifikátu.

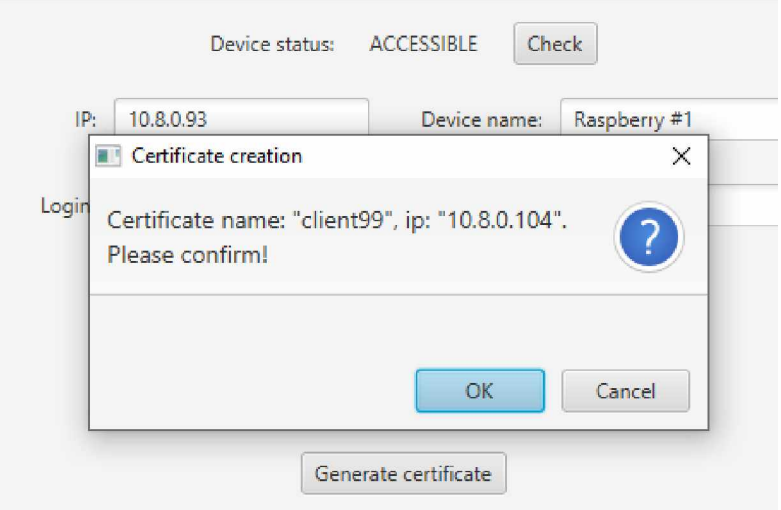
Hodnoty „Certificate name“ a „Certificate postfix“ spolu dohromady tvoří název certifikátu, pod kterým bude uložen na VPN serveru. Hodnota „Certificate postfix“ je po vytvoření certifikátu automaticky inkrementována, aby nedocházelo ke kolizi jmen. Cesta v poli „Server cert parts sources“ odkazuje na místo na VPN serveru, kde se po provedení první části generování certifikátu nacházejí části, ze kterých je výsledný certifikát sestaven.

Hodnota „Certificate name on client“ určuje jakým způsobem se bude jmenovat nově vytvořený certifikát na cílovém zařízení. Cesta v poli “Client target folder” pak určuje na jaké místo na cílovém zařízení se bude certifikát nahrávat.

Adresa v „Certificate ip“ určuje pod jakou IP adresou bude zařízení vystupovat v rámci VPN sítě. Do cesty nastavené v „Routing file target“ se na VPN server nahraje soubor s konfigurací směrování pro cílové zařízení. A výsledný certifikát se pak ještě uloží na VPN server do cesty určené v poli „Server target folder“.

4.5.2 Tvorba certifikátů

Samotný proces spuštění generování a nasazení certifikátu na zařízení se spouští z karty detailu daného zařízení pomocí tlačítka „Generate certificate“.



Obrázek 25: Dialog pro potvrzení generování certifikátu

Po kliknutí na dané tlačítko se zobrazí dialog (Obr. 25) pro překontrolování, na jaké nové adrese se v rámci VPN bude zařízení hlásit.

Obrázek 26: Průběh certifikačního procesu

Po spuštění certifikačního procesu se zobrazí textové pole (Obr. 26), ve kterém lze sledovat jeho průběh. Do tohoto pole se vypisují důležité informace o procesu a případné chyby, které během něj vzniknou.

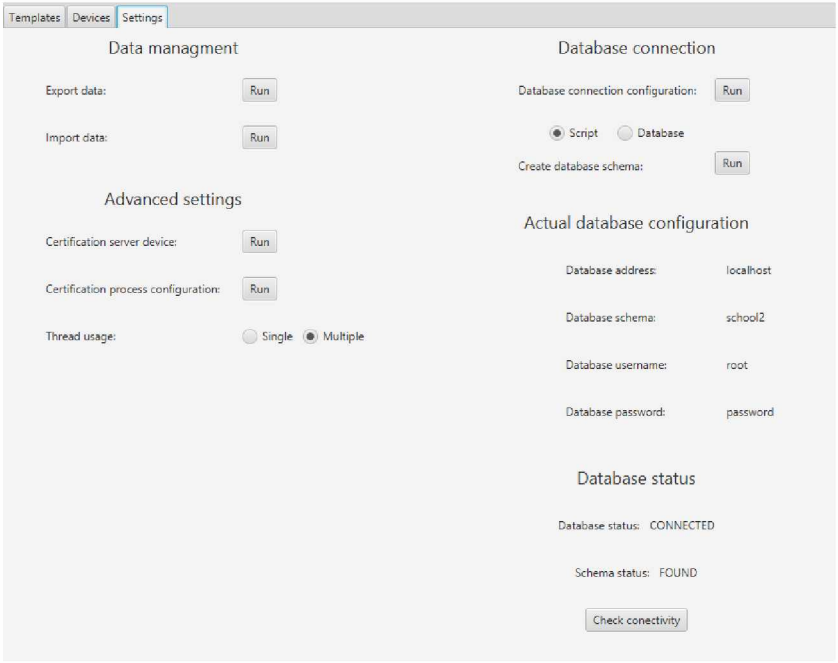
```
cd /etc/openvpn/easy-rsa;
. ./vars;
./build-key *CERT_NAME*;#Valid "Country Name";
 ; #Valid "Province Name";
 ; #Valid "Locality Name";
 ; #Valid "Organization Name";
 ; #Valid "Organizational Unit Name";
 ; #Valid "Common Name";
 ; #Valid "Name ";
 ; #Valid "Email Address ";
 ; #Valid "A challenge password";
 ; #Valid "An optional company name";
 ; #Valid "Sign the certificate?";
y; #Valid "certificate requests certified,"; #Invalid "failed to update database";
y;
```

Obrázek 27: Příkazy ke generování částí certifikátů

Certifikační proces využívá při generování částí certifikátů šablonu pro spuštění libovolné posloupnosti příkazů, která je popsána v předchozí podkapitole. Na obrázku 27 pak lze vidět, jakým způsobem je posloupnost příkazu pro vygenerování těchto částí zapsána.

4.6 Ostatní

4.6.1 Konfigurační část

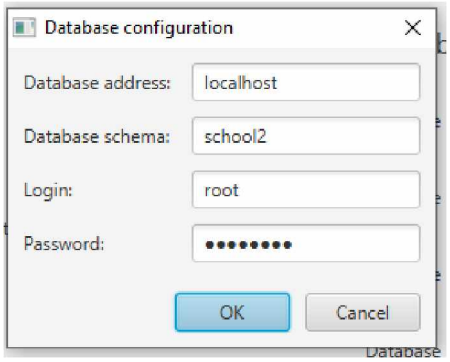


Obrázek 28: Karta „Settings“

Karta „Settings“ (Obr. 28) slouží pro nakonfigurování informací, které jsou nutné k běhu aplikace. Karta je rozdělena do tří sekcí. Sekce „Data managment“ bude popsána v podkapitole 4.6.1 „Import/Export dat”. Sekce „Advanced settings“ byla popsána během popisu konfigurace

VPN serveru, v podkapitole 4.5.1 „Konfigurační část“. V této sekci je dále možné nastavit způsob procesování z pohledu využívání vláken.

Poslední sekce se věnuje propojení s databází. Je rozdělena do tří podčástí. V podčásti „Actual database configuration“ lze vidět, jakým způsobem je aktualně nastavené propojení s databází. V podčásti „Database status“ lze pomocí tlačítka ověřit dostupnost databázového serveru a v něm pak také přímo dostupnost schématu.



Obrázek 29: Dialog pro připojení k databázovému serveru

V podčásti „Database connection“ je možné pomocí prvního tlačítka nakonfigurovat údaje sloužící k připojení do databázového serveru (Obr. 29). Druhým tlačítkem pak lze vygenerovat DDL SQL script pro tvorbu databázového schématu. Pomocí elementů „RadioButton“ se potom vybírá, zda má být tento script spuštěn přímo nad nastavenou databází, nebo vygenerován do souboru.

4.6.2 Import/Export dat

V kartě „Settings“ se nachází sekce „Data managment“, která slouží pro importování, nebo exportování dat z databáze. Data jsou čteny/ukládány do souborů ve formátu JSON.

5 ZÁVĚR

První kapitola byla zaměřena na pojem Internet věcí a jeho prvky – chytrá zařízení. Byly zde představeny nejdůležitější milníky v historii tohoto oboru a oblasti, ve kterých je využíván. Dále zde byla řešena otázka jeho bezpečnosti a rizika, která by neměla být podceňována.

Druhá kapitola pojednávala o virtuálních privátních sítích, principu jejich fungování a jejich výhodách a nevýhodách. Blíže zde byla představena OpenVPN, která je využívaná v prostředí, na které je praktická část bakalářské práce cílena.

Třetí kapitola představila přehled nejdůležitějších technologií, které jsou v rámci realizace praktické části práce využity.

Čtvrtá kapitola se pak věnuje samotnému návrhu a implementaci aplikace, což představuje praktickou část bakalářské práce. Je zde představeno prostředí, ve kterém se aplikace bude využívat a jakým způsobem se v něm dosavadně pracovalo. Dále je vysvětleno, jakým způsobem aplikace práci zefektivní. Poté následuje seznámení se všemi funkcemi aplikace doplněné o screenshoty. Podrobně jsou popsány všechny konfigurace potřebné pro správný chod aplikace, správa zařízení, správa šablon procesů a další funkce.

Hlavními požadavky na aplikaci bylo, aby dokázala zjednodušit certifikační proces pro jednotlivá zařízení, potřebný pro připojení zařízení do VPN a zjednodušit výměnu softwarových balíčků, tak aby se dala vykonávat z aplikace pro více zařízení najednou. Oba procesy byly dosavadně vykonávány manuálně. Aplikace zajišťuje vykonávání obou procesů plně automaticky, přímo z aplikace. Pro zjednodušení výměny softwarových balíčků byla přidána možnost paralelního procesování nad vybranými zařízeními.

Hlavní požadavky byly splněny. Nad rámec požadavků byla přidána funkce, která dokáže nad zařízeními vykonávat libovolnou posloupnost příkazů, s možností řídit tento proces podmíněnými výsledky jednotlivých příkazů.

Cíle bakalářské práce byly splněny v plném rozsahu.

POUŽITÁ LITERATURA

[1] POHANKA, Pavel. Internet věcí. Pavel Pohanka [online]. 2017 [cit. 2020-04-16]. Dostupné z: http://www.pavelpohanka.cz/internet-of-things/

[2] Internet of Things (IoT) History. Postscapes [online]. 11. 12. 2019 [cit. 2020-04-19]. Dostupné z: https://www.postscapes.com/iot-history/

[3] Internet of Things: Status and implications of an increasingly connected world. United States Government Accountability Office [online]. Květen 2017, s. 4 [cit. 2020-04-16]. Dostupné z: https://www.gao.gov/assets/690/684590.pdf

[4] SILVERIO, Manuel. What is a smart device? — The key concept of the Internet of Things. towards data science [online]. 29. 12. 2019 [cit. 2020-04-19]. Dostupné z: https://towardsdatascience.com/what-is-a-smart-device-the-key-concept-of-the-internet-of-things-52da69f6f91b

[5] TEICHER, Jordan. The little-known story of the first IoT device. IBM [online]. 7. 2. 2018 [cit. 2020-04-19]. Dostupné z: https://www.ibm.com/blogs/industries/little-known-story-first-iot-device/

[6] CAMERON, Lori. Internet of Things Meets the Military and Battlefield: Connecting Gear and Biometric Wearables for an IoMT and IoBT. IEEE Computer Society [online]. 2016 [cit. 2020-04-17]. Dostupné z: https://www.computer.org/publications/tech-news/research/internet-of-military-battlefield-things-iomt-iobt

[7] ČIČVÁKOVÁ, Michala. PRŮMYSL 4.0 A JEHO VLIV NA SVĚT PRÁCE. Národní ústav pro vzdělávání [online]. 2017 [cit. 2020-04-19]. Dostupné z: http://www.nuv.cz/vystupy/cast-1-prumysl-4-0-a-jeho-vliv-na-svet-prace

[8] VOJÁČEK, Antonín. Co se skrývá pod výrazy Industry 4.0 / Průmysl 4.0 ? Automatizace.hw.cz: rady a poslední novinky z oboru [online]. 19. 3. 2016 [cit. 2020-04-19]. Dostupné z: https://automatizace.hw.cz/mimochodem/co-je-se-skryva-pod-vyrazy-industry-40-prumysl-40.html

[9] LOM, Michal a Ondřej PŘIBYL. Rizika chytrých zařízení a jejich zabezpečení. Tzbinfo [online]. 3.4.2017 [cit. 2020-04-19]. Dostupné z: https://elektro.tzb-info.cz/inteligentni-budovy/15569-rizika-chytrych-zarizeni-a-jejich-zabezpeceni

[10] SVOBODA, Ivan. Podrobně: (Ne)bezpečnost internetu věcí. BusinessIT [online]. 5. 2. 2018 [cit. 2020-04-19]. Dostupné z: http://www.businessit.cz/cz/podrobne-ne-bezpecnost-internetu-veci.php

[11] ŽIDKOVÁ, Nikola. Bezpečnost komunikačních technologií IoT [online]. Praha, 2017 [cit. 2020-04-19]. Dostupné z: https://theses.cz/id/js4ujk/. Diplomová práce. Vysoká škola ekonomická v Praze. Vedoucí práce Petr Doucek.

[12] MATHEWS, Lee. Criminals Hacked A Fish Tank To Steal Data From A Casino. Forbes [online]. 27. 7. 2017 [cit. 2020-04-19]. Dostupné z: https://www.forbes.com/sites/leemathews/2017/07/27/criminals-hacked-a-fish-tank-to-steal-data-from-a-casino/#e9c28d532b96

[13] KUPKA, Michael. (Ne)bezpečnost IoT aneb Stále podceňované riziko. SystemOnLine [online]. 2. 9. 2019 [cit. 2020-04-19]. Dostupné z: http://m.systemonline.cz/it-security/nebezpecnost-iot-aneb-stale-podcenovane-riziko.htm

[14] Java. TechTerm [online]. 19. 4. 2012 [cit. 2020-04-19]. Dostupné z: https://techterms.com/definition/java

[15] ŠTECHMÜLLER, Petr. Úvod do JavaFX. ITnetwork.cz [online]. 14. 3. 2019 [cit. 2020-04-19]. Dostupné z: https://www.itnetwork.cz/java/formulare/javafx/uvod-do-javafx

[16] PEREDA, José. Home. Github [online]. 28. 5. 2018 [cit. 2020-04-19]. Dostupné z: https://github.com/gluonhq/scenebuilder/wiki

[17] PETEROVÁ, Alena. Seznámení s Hibernate ORM. BCV solutions [online]. 16. 4. 2014 [cit. 2020-04-19]. Dostupné z: https://blog.bcvsolutions.eu/seznameni-s-hibernate-orm/

[18] DWIKA, Herawan. What is MySQL: MySQL Explained For Beginners. Hostinger tutorials [online]. 4. 3. 2020 [cit. 2020-04-19]. Dostupné z: https://www.hostinger.com/tutorials/what-is-mysql

[19] JSch - Java Secure Channel. JCraft [online]. 18. 3. 2015 [cit. 2020-04-19]. Dostupné z: http://www.jcraft.com/jsch/

[20] ROUSE, Margaret. Secure Shell (SSH). TechTarget: SearchSecurity [online]. 17. 3. 2016 [cit. 2020-04-21]. Dostupné z: https://searchsecurity.techtarget.com/definition/Secure-Shell

[21] VERGES, Chris. Expect4J. GitHub [online]. 22. 10. 2013 [cit. 2020-04-21]. Dostupné z: https://github.com/Alexey1Gavrilov/ExpectIt

[22] ZHENG, Mary. Logback Tutorial for Beginners. Java Code Geeks [online]. 27. 7. 2018 [cit. 2020-04-21]. Dostupné z: https://examples.javacodegeeks.com/enterprise-java/logback/logback-tutorial-beginners/

[23] HORDĚJČUK, Vojtěch. Maven. Voho [online]. 2016 [cit. 2020-04-21]. Dostupné z: http://voho.eu/wiki/maven/

[24] BOŘÁNEK, Roman. VPN pro začátečníky: princip fungování, výhody a nevýhody. ROOT.CZ [online]. 7. 7. 2017 [cit. 2020-04-22]. Dostupné z: https://www.root.cz/clanky/vpn-pro-zacatecniky-princip-fungovani-vyhody-a-nevyhody/

[25] EMPEY, Charlotte Empey. Jak funguje VPN a proč se vyplatí ji mít. Avast: blog [online]. 13. 3. 2018 [cit. 2020-04-22]. Dostupné z: https://blog.avast.com/cs/jak-funguje-vpn-a-proc-se-vyplati-ji-mit

[26] LOW, Jerry. Virtual Private Network (VPN): A Very Detailed Guide for Newbies. Web Hosting Secret Revealed [online]. 23. 3. 2020 [cit. 2020-04-22]. Dostupné z: https://www.webhostingsecretrevealed.net/the-a-to-z-vpn-guide/