

# **System pro logování a monitoring stavu kotlů na pevná paliva**

System for Logging and Monitoring of Boilers

Bc. Petr Štípek

---

Diplomová práce  
2014



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2013/2014

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Petr Štípek**  
Osobní číslo: **A12397**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
Forma studia: **prezenční**

Téma práce: **Systém pro logování a monitoring stavu kotlů na pevná paliva**

Zásady pro vypracování:

1. Analyzujte požadavky na aplikaci pro logování a monitoring stavu kotlů Ponast, které jsou řízeny systémem Sigmatek.
2. Prozkoumejte možnosti řídicího systému Sigmatek, vývojového prostředí LASAL a knihoven pro komunikaci přes TCP/IP.
3. Navrhněte a implementujte vhodný komunikační protokol pro logování a monitoring stavu kotlů jak na straně systému Sigmatek, tak na straně web serveru.
4. Prozkoumejte dostupné open source web aplikace pro monitoring a logování stavu zařízení, připojených do sítě TCP/IP a zjistěte, zda by bylo možné využít některou z nich pro splnění požadavků zadavatele, např. implementací nového modulu.
5. Implementujte web aplikaci popř. nový modul pro logování a monitoring kotlů.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. KABIR, M. Apache server 2. Vyd. 1. Brno: Computer Press, 2004, 724 s. ISBN 80-251-0319-6.
2. GUTMANS, Andi, Stig Sther BAKKEN a Derick RETHANS. Mistrovství v PHP 5. Vyd. 1. Překlad Bogdan Kiszka. Brno: CP Books, 2005, 655 s. ISBN 80-251-0799-X.
3. CCB SPOL. S.R.O. SystemOnLine: S přehledem ve světě informačních technologií [online]. 2014 [cit. 2014-01-09]. Dostupné z: <http://systemonline.cz/>
4. ORACLE CORPORATION. MySQL: The world's most popular open source database [online]. 2014 [cit. 2014-01-09]. Dostupné z: <http://mysql.com/>
5. SIGMATEK GMBH & CO KG. Sigmatek: Complete automation solutions for machine building [online]. 2014 [cit. 2014-01-09]. Dostupné z: <http://sigmatek-automation.us/>
6. DULÍK, Tomáš, Jan DUBINA, Ondřej FIBICH, Michal KLIMENT a David RAŠKA. FreenetIS: Informační systém pro neziskové sítě [online]. 2014 [cit. 2014-01-09]. Dostupné z: <http://freenetis.org/>

Vedoucí diplomové práce:

**Ing. Tomáš Dulík, Ph.D.**

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

**21. února 2014**

Termín odevzdání diplomové práce:

**20. května 2014**

Ve Zlíně dne 21. února 2014

  
prof. Ing. Vladimír Vašek, CSc.  
*děkan*



  
doc. Mgr. Roman Jašek, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Tato práce se zabývá technologií, jejímž cílem je zlepšování služeb společnosti PONAST spol. s r.o. jejich zákazníkům. Zejména se jedná o urychlení reakce zákaznické podpory na případné chyby a odchylky v systému kotlů na tuhá paliva, aniž by bylo potřeba iniciativy ze strany zákazníka. Budou zde zmíněny nástroje a možnosti řešení vzdálené komunikace mezi produkty společnosti a jejich správci. Zmíní se i komplexní produkt Sigmatek LASAL, který je použit jako operační systém kotlů na tuhá paliva. V praktické části bude navrženo řešení pro vzdálené monitorování chování, změn a chyb koncových zařízení postavené na již existujícím projektu Freenetis. Výsledkem práce je především zdokumentování návrhu a postupu realizace rozšíření pro již existující webovou aplikaci Freenetis.

Klíčová slova: Monitoring vzdálených zařízení, kotle na tuhá paliva, správa topologie sítě

## **ABSTRACT**

This thesis deals with a technology used for improvement of services provided by PONAST Ltd to their customers. In particular, improving the monitoring and customer support of their solid fuel boilers without the need for an initiative from the customer. The work elaborates possible solutions of remote communication between company products and their management. It also refers to Sigmatek LASAL, which is used as the operating system of PONAST solid fuel boilers. In the practical part, there is a proposed solution for remote monitoring, changes and errors of boilers, based on an existing project Freenetis. The result documents the concept and implementation of new Freenetis extension.

Keywords: Monitoring of remote devices, solid fuel boilers, management of the network topology

Děkuji vedoucímu práce panu Ing. Tomáši Dulíkovi Ph.D. za pomoc při zpracování této diplomové práce a jeho cenné rady. Dále bych chtěl poděkovat Ing. Romanu Červeňanovi za pomoc se zpracováním kódů pro kotle na tuhá paliva.

**Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD .....</b>	<b>9</b>
<b>I TEORETICKÁ ČÁST .....</b>	<b>10</b>
<b>1 ZÁMĚR A SPECIFIKACE ZAŘÍZENÍ .....</b>	<b>11</b>
1.1 VÝBĚR ZAŘÍZENÍ SE ZOHLEDNĚNÍM NA PROSTORY .....	11
1.1.1 Výkon kotle a účinnost spalování.....	11
1.2 PROBLEMATIKA PARALELIZACE .....	13
1.3 DISTRIBUOVANÉ ŘEŠENÍ.....	13
1.4 SLEDOVÁNÍ STAVŮ ZAŘÍZENÍ .....	14
<b>2 POŽADAVKY NA APLIKACI .....</b>	<b>15</b>
2.1 SPECIFIKACE PRO LOGOVÁNÍ A MONITOROVÁNÍ.....	15
2.1.1 Číselné hodnoty .....	16
2.1.2 Stavové hodnoty .....	16
2.1.3 Zprávy o události.....	17
2.2 KOMUNIKACE MEZI ZAŘÍZENÍM A APLIKACÍ.....	17
<b>3 SIGMATEK LASAL.....</b>	<b>18</b>
<b>4 MONITOROVACÍ SYSTÉMY .....</b>	<b>19</b>
4.1 FREENETIS .....	20
4.1.1 Zabezpečení a uživatelské účty .....	20
4.1.2 Zařízení a nástroje .....	21
4.1.3 Instalační balíčky.....	21
4.1.3.1 Monitoring .....	21
<b>II PRAKTICKÁ ČÁST .....</b>	<b>22</b>
<b>5 PŘENOS DAT Z KOTLE.....</b>	<b>23</b>
<b>6 IMPLEMENTACE A DOSTUPNOST SYSTÉMU FREENETIS .....</b>	<b>27</b>
<b>7 KOMUNIKAČNÍ ROZHRANÍ.....</b>	<b>29</b>
7.1 SOUČASNÁ IMPLEMENTACE .....	29
7.2 INOVACE SOUČASNÉ IMPLEMENTACE .....	30
7.3 PŘÍJEM DAT.....	32
7.3.1 Skript pro zpracování.....	32
7.3.2 Autorizace .....	33
7.3.3 Identifikace nových typů událostí .....	33
<b>8 VLASTNÍ ŘEŠENÍ APLIKACE .....</b>	<b>34</b>
8.1 DATABÁZOVÁ STRUKTURA A APLIKAČNÍ LOGIKA .....	35
8.1.1 Tabulka zařízení .....	36
8.1.2 Události a parametry.....	36
8.1.2.1 Využití serializace a deserializace.....	37
8.1.3 Grafy .....	39
8.1.4 Rozšíření výchozích struktur.....	40
8.2 UŽIVATELSKÉ ROZHRANÍ.....	41
8.2.1 Přidání nového zařízení .....	42
8.2.2 Nastavení přístupových práv.....	45
8.2.3 Seznam událostí.....	46

---

8.2.4	Seznam parametrů .....	46
8.2.5	Nastavení a zobrazování grafů .....	47
8.2.6	Detaily událostí zařízení s vizualizací logovaných parametrů.....	49
<b>ZÁVĚR .....</b>		<b>52</b>
<b>ZÁVĚR V ANGLIČTINĚ.....</b>		<b>53</b>
<b>SEZNAM POUŽITÉ LITERATURY .....</b>		<b>54</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>		<b>55</b>
<b>SEZNAM OBRÁZKŮ.....</b>		<b>56</b>
<b>SEZNAM TABULEK .....</b>		<b>57</b>
<b>SEZNAM PŘÍLOH.....</b>		<b>58</b>

## ÚVOD

Toto téma mě zaujalo zejména z důvodu mého zájmu o automatizaci a optimalizaci provozu zařízení, která komunikují pomocí TCP/IP protokolu a mohou tak být vzdáleně monitorovány nebo řízeny, ať už v rámci firemního intranetu nebo Internetu. Zkušenosti v této větví informačních technologií tak pomohou zlepšit vlastní znalosti a dovednosti, ale také mohou přinést reference potřebné pro další vývoj a spolupráci s podniky, jejichž požadavky na webové aplikace směřují právě tímto směrem a jejichž počet se v posledních několika letech stále zvyšuje.

Cílem této práce je rozšíření některé z již existujících webových aplikací. Rozšíření se týká zejména příjmu dat z kotlů na tuhá paliva, které mohou posílat informace o změně svého stavu pomocí událostí a parametrů snímaných za provozu. Kotle jsou programovány za pomoci systému Sigmatek LASAL. Software kotlů je již připraven pro testovací účely, ale jeho vývoj stále neskončil. Při návrhu webové aplikace se tak musí brát ohled na již existující řešení, ale i možnost rozvoje v rámci zlepšení monitorování provozu.

V této práci bych chtěl nastínit problematiku monitorování kotlů na tuhá paliva, které jsou paralelizovány v rámci jedné otopné soustavy a přiblížit tak nároky, které vznikají na webovou aplikaci. Bude zde zmíněn produkt Sigmatek LASAL a webové projekty umožňující monitorování síťových prvků. Na jednom z těchto webových projektů s otevřeným zdrojovým kódem bude navržena praktická část této práce, tj. struktura aplikace, databáze a aplikační logika včetně vysvětlení použitých metodik zpracování dat. Závěr praktické části bude věnován nastavením a popisu práce s vytvořeným rozšířením pro webovou aplikaci. Tímto by se měla čtenáři ukázat integrace do již existujícího systému.

## **I. TEORETICKÁ ČÁST**

## 1 ZÁMĚR A SPECIFIKACE ZAŘÍZENÍ

Firma Ponast spol. s r. o. se zabývá vývojem kotlů na tuhá paliva. V rámci vývoje nové řady zařízení a řešení problému paralelizace, vytíženosti a údržby těchto zařízení potřebují navrhnout systém, který bude shromažďovat informace o stavech jednotlivých kotlů. K těmto údajům by se poté měli dostat jak technici, tak majitelé zařízení a to z důvodu předcházení výpadkům a poruchám.

### 1.1 Výběr zařízení se zohledněním na prostory

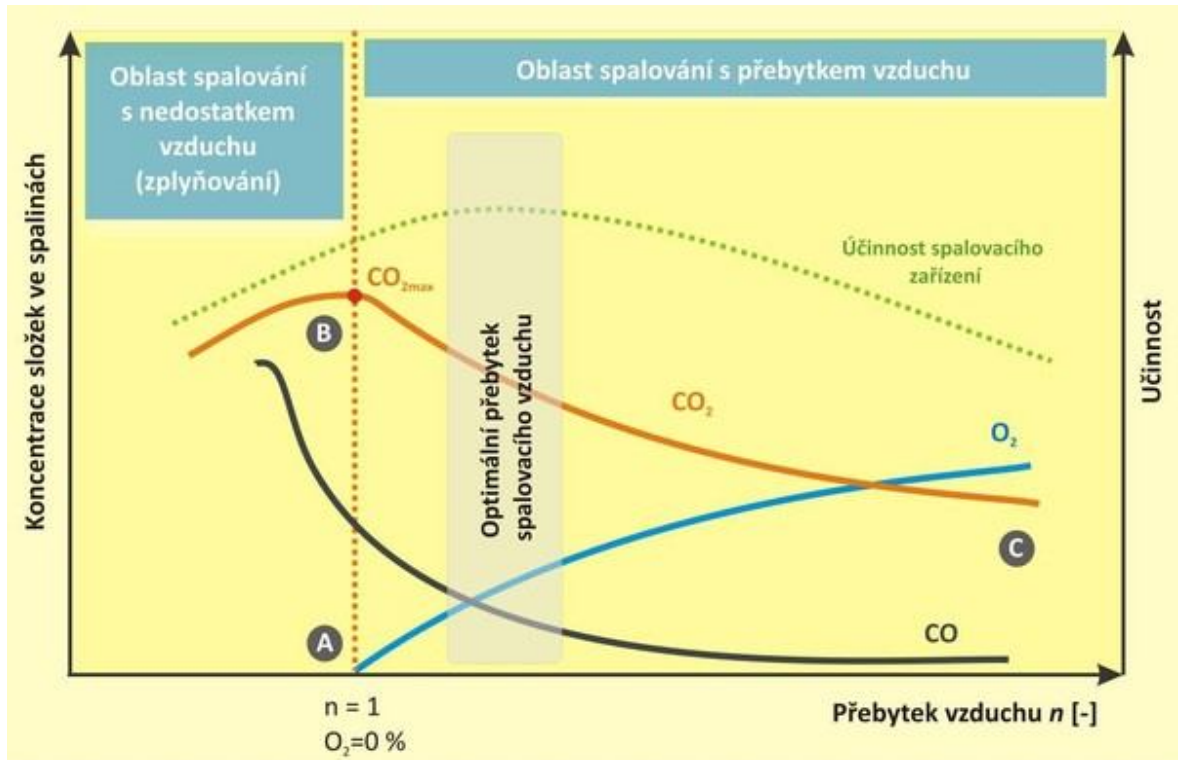
Kotle pro vytápění prostor se vybírají právě se zohledněním na typ prostor a jejich velikost. Může se zdát, že čím větší prostory chceme vytápět, tím bude výhodnější instalovat výkonnější kotel. Ale právě kotle na tuhá paliva mají tu nevýhodu, že v případě, kdy nechceme prostory vytápět, ale už jen temperovat, jsme donuceni topit jen na zlomkový výkon kotle. V takovém případě rapidně klesá účinnost spalování a je na zvážení, zda by nebylo lepší místo jednoho výkonného kotle zvolit dva méně výkonné a roztápět je podle potřeby.

Proces roztápění se dá zejména v průmyslových řešeních automatizovat, a to tak že se topí peletami, které jsou přikládány automatickým podavačem a řídicí systém spolu s kotlem je tak schopen roztopit i doposud studený kotel.

#### 1.1.1 Výkon kotle a účinnost spalování

Výkon kotle na tuhá paliva je závislý na několika okolnostech. Prvním je výkon topeniště udávaný přímo výrobcem daného kotle, značící maximální možný výkon v závislosti na optimálním množství spalovacího vzduchu pro hoření a odvod spalin. Spalovací rovnice, které bilančně popisují reakci mezi hořlavinou a kyslíkem, zjednodušeně vyjadřují množství kyslíku (vzduchu), který je potřeba k tomu, aby došlo k jejich reakci. Toto množství nazýváme stechiometrické (teoretické) a představuje minimální potřebu pro spalování při reakci veškerého materiálu a za předpokladu, že se navzájem sejdou všechny reakční látky. V reálném hoření tomu tak není, proto je množství spalovacího vzduchu mnohem větší, než by bylo teoreticky potřebné, čímž se zvyšuje dokonalost spalování, snižuje se obsah CO a zvyšuje se tak účinnost. Přebytek spalovacího vzduchu  $n$  se pohybuje v poměru 1,5 až 2 pro automatické kotle, 2 až 4 pro kotle a kamna s ručním přikládáním paliva. Se zvyšujícím se přebytkem spalovacího vzduchu ovšem postupně klesá teplota v ohništi, takže začne klesat i kvalita spalování, dále roste množství

spalin, tím také roste komínová ztráta a snižuje se účinnost spalovacího zařízení. Každé zařízení má svůj optimální přebytek spalovacího vzduchu, při kterém je dosaženo optimálních hodnot koncentrace CO a účinnosti viz Obrázek 1.



Obrázek 1 Graf závislosti koncentrace složek ve spalinách a účinnosti na množství vzduchu při spalování pevných paliv [12]

Při reálném provozu měníme přebytek spalovacího vzduchu tím, jak regulujeme množství nasávaného spalovacího vzduchu. Takže pokud otevřeme dvířka popelníku, regulační klapky vzduchů apod., budeme se pohybovat v pravé části grafu. Pokud uzavřeme přívody spalovacích vzduchů, začne se snižovat přebytek spalovacího vzduchu a v grafu se přemísťujeme směrem vlevo. Jak moc se posuneme vlevo, záleží také na tom, jak dobře je spalovací zařízení těsné (těsnicí šňůry apod.). Pokud uzavřeme všechny přívody vzduchu a nepozorujeme evidentní pohasnutí plamene (např. u krbových kamen to můžeme pozorovat přes sklo dvířek), tak se spalovací vzduch dostává k palivu netěsnostmi a my se stále pohybujeme v pravé části grafu, což je doprovázeno vysokou komínovou ztrátou, a tedy nízkou účinností. Pokud je zařízení těsné, můžeme se po uzavření přívodu vzduchu dostat až do oblasti, kde je nedostatečné množství spalovacího vzduchu a vlastně začneme palivo zplyňovat. Toto se velmi často děje, pokud na noc přiložíme velkou dávku paliva a pak, protože jdeme již spát, všechny přívody vzduchů uzavřeme. Je pravda, že se podstatně prodlouží doba hoření (zplyňování), ale křivka CO nám ukazuje, k jak

dramatickému nárůstu produktů nedokonalého spalování dojde a také účinnost je velmi špatná. [12]

Křivka označená jako „O2“ ukazuje průběh koncentrace kyslíku v závislosti na přebytku spalovacího vzduchu. Pokud bychom k palivu přivedli právě tolik vzduchu, kolik je třeba (přebytek  $n = 1$ ), veškerý by se spotřeboval a jeho koncentrace by byla nulová, jak jde vidět v bodě „A“. Při tomto stavu by veškerý uhlík shořel na CO<sub>2</sub>, takže jeho koncentrace by byla maximální, tehdy hovoříme o  $CO_{2max}$ , viz bod „B“. Se zvyšujícím se přebytkem spalovacího vzduchu poroste koncentrace kyslíku ve spalinách. Naopak koncentrace CO<sub>2</sub> bude klesat, protože přebytečný vzduch bude CO<sub>2</sub> ředit. Se vzrůstajícím přebytkem spalovacího vzduchu se zlepšuje spalování, klesá koncentrace CO, ale na druhou stranu roste množství spalin, takže roste množství tepla vyvedeného komínem, čímž klesá účinnost.

Pro správné nastavení hoření a tím i účinnosti je tak potřeba najít kompromis u koncentrace vzduchu a emisí CO. Každé spalovací zařízení pro dané palivo má vlastní optimální hodnotu spalovacího vzduchu (viz plocha v grafu).

## 1.2 Problematika paralelizace

Pokud zvolíme řešení, kdy pro vytápění využíváme několika paralelních kotlů spínaných podle potřeby, dostaneme se do problému nesouměrného opotřebení zařízení. V případě běžných řešení se využívá jednoho kotle typu master a ostatních typu slave. Master má v sobě implementovanou logiku pro vlastní řízení ale i pro spouštění podřadných kotlů v případě potřeby většího výkonu.

Problémem tohoto návrhu je vysoké vytížení hlavního kotle, který je v provozu mnohonásobně delší dobu než ostatní kotle a narůstá tak riziko jeho výpadku případě vysokého opotřebení. Pokud by takový stav nastal a master kotel by vypadl, tak by automaticky přestaly fungovat i ostatní kotle a celá soustava by byla takto odstavena až do výměny hlavního kotle.

## 1.3 Distribuované řešení

Předcházet celkovému výpadku se dá pouze tak, že se zavede distribuované přidělování stavu kotle typu master a to buď podle počtu otopných hodin, nebo v případě

výpadku dosavadního hlavního kotle. Jednotlivá zařízení je tak nutné zapojit do hvězdicové topologie a musí být mezi nimi navrženo rozhraní, kterým budou spolu komunikovat a předávat si tak vedoucí roli.

V případě výpadku v takto budované topologii se pouze odstaví nefunkční kotel, který je buď zastoupen jiným zařízením, nebo celá soustava topí s nižším celkovým výkonem, ale nedojde k úplnému výpadku.

#### **1.4 Sledování stavů zařízení**

Distribuované řešení zmírňuje dopady výpadku zařízení, ale není schopné výpadku předcházet. Jelikož zařízení mezi sebou tvoří hvězdicovou topologii, dalo by se využít síť Ethernet pro komunikaci mezi jednotlivými zařízeními a zároveň pomocí autorizace MAC a IP adresou by zařízení mohly posílat informace o svém stavu na vzdálený server, kde by měla přístup údržba. Takto může jeden člověk hlídat velké množství zařízení a také může provádět analýzy z provozu, vytvářet návrhy pro zlepšení nebo v případě blížící se doby údržby vyslat technika k danému zařízení dříve než dojde k poruše.

## 2 POŽADAVKY NA APLIKACI

Celý systém by měl sloužit jak výrobci zařízení, tak i jejich uživatelům. Je třeba, aby systém umožňoval přístup několika druhům uživatelů:

- **Administrátor:** Hlavní správce informačního systému.
- **Správce/Technik:** Podpora ze strany dodavatele, která bude dohlížet na provoz zařízení
- **Zákazník:** Uživatel, který bude mít přehled o svých zařízeních a jejich provozu

Objektem správy budou zařízení organizované v sítích intranetů s protokolem TCP-IP. Jednotlivá zařízení budou organizovány do topologií a je nutné pro jejich identifikaci používat IP a MAC adres. Je potřeba rozlišit několik základních druhů zařízení:

- **Přístupový bod**
- **Router**
- **Switch**
- **Kotel**

Pomocí těchto zařízení jsme schopni vytvářet sítě, podsítě a je možné přistupovat k nim i ze sítě Internet, přes kterou budou kotle na tuhá paliva odesílat informace o svém stavu. Zároveň bude ověřována funkčnost a provozuschopnost těchto zařízení. Je možné použít i další zařízení jako jsou PC, notebooky nebo jiné jednotky připojitelné přes rozhraní Ethernet a Wi-Fi pro budoucí využití.

Dále je potřeba, aby byl implementován finanční systém. Každý uživatel by měl mít vlastní účet a systém by měl disponovat funkcionalitou pro vedení poplatků za provedení prací, ale i pravidelné příspěvky, které mohou mít formu tarifu. Bylo by vhodné implementovat algoritmy umožňující propojení systému s bankou nebo účetním systémem a automatizovat tak bankovní převody.

### 2.1 Specifikace pro logování a monitorování

Každé zařízení, pokoušející se spojit se serverem musí projít autorizací, kdy se ověří IP adresa podsítě zařízení a jeho MAC adresa s daty u zařízení uložených v systému. Po úspěšné autorizaci poté budou parametry zařízení převedeny do vhodného formátu a archivovány spolu s časovým razítkem získaným ze zařízení. Rozhraní pro příjem dat musí

být navrženo s ohledem na výkon serveru a dostatečnou propustností pro logování stavů desítek zařízení v pěti minutových intervalech.

Jelikož zařízení, která budou monitorována, jsou stále ve vývoji, tak není přesně známo, kolik parametrů a jaké bude potřeba logovat, ale jejich počet by měl být řádově v desítkách až stovkách. Je tedy potřeba navrhnout pružné rozhraní umožňující další rozšiřování do budoucna, aniž by bylo nutné zasahovat do programu aplikace.

Data, přijímaná a ukládaná do aplikace jsou trojího typu:

- 1 Číselné hodnoty**
- 2 Stavové hodnoty**
- 3 Zprávy o události**

### **2.1.1 Číselné hodnoty**

Toto bude nejspíše nejpočetnější skupina parametrů, které se budou zaznamenávat. Jedná se zejména o hodnoty z teploměrů, průtoky topné vody, množství spáleného materiálu, výsledky výpočtů jako jsou výkony nebo účinnosti. Bylo by vhodné tyto parametry zobrazovat s časovou osou tak, aby byly vidět změny těchto parametrů. Některé hodnoty (jako jsou teplota v budově a mimo ní) mohou být svázané s jinými, podle kterých se na bázi ekvitermního vytápění reguluje teplota topné vody. U těchto hodnot by měla být možnost shlukování do grafů, kde by byly viditelné jejich závislosti a průběhy.

### **2.1.2 Stavové hodnoty**

Ke kotli mohou být připojeny komponenty, jako jsou motory, relé, optické závory, čerpadla, vzduchotechnika a spousta dalších, jejichž činnost se dá vyjádřit formou stavů. Motor běží nebo stojí, závora je otevřena nebo zavřena. Ale kromě dvoustavových zde mohou být i více-stavové komponenty, jako je třeba zmíněná vzduchotechnika. Motor rozhánějící vzduch do místností se může regulovat spojitě, kdy se mění počet otáček, nebo se mění rychlostní stupně.

Stavy jako takové mohou být reprezentovány poměrně jasným značením číselnou hodnotou, značící číslo stupně. Mohou být typu pravda/nepravda (otevřeno / zavřeno), nebo také mohou být pro uživatele zcela nesmyslného charakteru jako je binární nebo hexadecimální číslo, či zkratka používaná uvnitř systému.

Pro zobrazení stavových hodnot je nutné nejprve provést překlad hodnoty na srozumitelnou formu a tento překlad by měl být využíván vždy jen v rámci jednoho parametru (proměnné), aby nedocházelo ke kolizím nebo nesprávným překladům.

### 2.1.3 Zprávy o události

V závislosti na vnitřním stavu zařízení může dojít k situaci, kdy je potřeba upozornit obsluhu nebo správce, aby na základě upozornění vykonal příslušné kroky. Těmi mohou být například potřeba provedení údržby zařízení, pokud je v provozu příliš dlouho a zvyšuje se riziko výpadku. Dalším důvodem naléhavé události může být vyčerpání dostupných zásob paliva, kdy následkem by byl opět výpadek. Zprávy je potřeba zaznamenávat tak, aby k nim byl rychlý přístup obsluhy, nebo aby obsluha byla sama upozorněna na novou událost.

## 2.2 Komunikace mezi zařízením a aplikací

Zařízení a webová aplikace budou spolu komunikovat pomocí HTTP požadavků, ve kterých se bude přenášet IP a MAC adresa zařízení pro spárování a autorizaci, dále časové razítko pro zachování informace o čase zaznamenání údajů a samotná data.

Komunikace pomocí HTTP požadavků není nijak chráněná, pokud by někdo chtěl vysílat falešná data, postačí mu znát kombinaci IP a MAC adresy. Bylo by vhodné tuto komunikaci zabezpečit, nebo alespoň ošetřit již zapsaná data proti jejich přepsání.

### 3 SIGMATEK LASAL

Rakouská firma Sigmatek se věnuje návrhu a výrobě automatizačních systémů pro stroje a výrobní zařízení s dlouholetou zkušeností na trhu již od roku 1988. Zabývají se vývojem jak hardwarových, tak i softwarových komponent s širokou škálou využití. [4]

LASAL je objektově orientované programové prostředí s technologií klient-server. Jeho hlavní předností je grafické znázornění tříd, funkcí a přenosů dat pro snadnější rozpoznání a změnu složitých systémových vztahů. Nabízí několik funkcí pro vytváření řešení automatizace: [4]

- **LASAL CLASS** - objektově orientované programování
- **LASAL SCREEN** - vizualizace na grafickém terminálu
- **LASAL MOTION** – správa přenosů a pohybových komponent
- **LASAL SAFETY** - bezpečnost programování a konfigurace
- **LASAL SERVICE** - servis a údržba na dálku přes internet

Hlavní předností pro tento projekt je možnost komunikace pomocí TCP/IP protokolu pomocí HTTP požadavků. V manuálu tuto problematiku můžeme nalézt jako kapitulu „LASAL HTTP Client“, kde je popsána komunikace a předávání dat pomocí GET a POST metod HTTP protokolu. Při srovnání s jinými webovými klienty zde nejsou omezující parametry. Jediným problémem může být nedostatek operační paměti při skládání jednotlivých požadavků v závislosti na celém vyvíjeném systému. [5]

K zabezpečení komunikace mezi zařízením Sigmatek a serverem je možno využít zabezpečení šifrováním komunikace pomocí vrstvy SSL a HTTPS protokolu. Součástí aplikace jsou i certifikáty, které si sami můžeme vytvářet. Je pouze nutné definovat certifikační autoritu, u které se ověřuje platnost certifikátu během zpracování požadavku. Výsledkem takového požadavku je i informace o zabezpečení odpovědi přijaté ze strany serveru. [5]

## 4 MONITOROVACÍ SYSTÉMY

Na Internetu se můžeme dočíst o celé řadě monitorovacích systémů zaměřených na síťová zařízení, jejich správu a vyhodnocování výkonu síťového provozu. Tyto systémy jsou vysoce specializovány a optimalizovány pro maximální výkon a správu pokud možno co největšího počtu síťových zařízení. Mezi nimi se najdou i kvalitní systémy s otevřeným zdrojovým kódem nabízející celou řadu nástrojů. Stojí za zmínění systémy Zabbix, Nagios a Freenetis. Všechny tři systémy jsou navrženy a optimalizovány k provozu na operačním systému Linux se správou pomocí webového prostředí a nabízejí podobné nástroje.

Zabbix slouží k monitorování aktivních síťových prvků (PC, servery, tiskárny, modemy, switche, UPS,...), které jsou připojeny do počítačové sítě. Je tedy možné říci, že můžeme sledovat stav a sbírat různé informace o všem, co má IP adresu. Metody pro sledování a zjišťování informací jsou různé. Počínaje jednoduchým ICMP echo request (ping) přes použití složitějších metod SNMP (Simple Network Management Protocol), IPMI (Intelligent Platform Management Interface), JMX (Java Management Extensions). Je možné použít také k monitoringu SSH/Telnet anebo použití agenta, který je dostupný pro většinu dnes používaných operačních systémů. Při použití agenta je možné monitorovat informace o stavu hardware (operační paměť, procesor, úložné zařízení,...), ale také systémové informace a stav běžících služeb. V neposlední řadě je možné integrovat do prostředí vlastní externí skripty, nebo využít API a vytvořit si tak vlastní testy. Pomocí API lze také komunikovat s jinými nástroji. Co se týká počtu dohlížených zařízení, dle tvůrců, je možné monitorovat přes 100 000 hostů a provádět tak milion vyhodnocení za minutu, což je pochopitelně závislé na systémových zdrojích serveru, na kterém je dohledový systém provozovaný. Zabbix může pracovat distribuovaně, což znamená, že v různých vzdálených lokalitách běží Zabbix v režimu proxy a data se následně přenáší na centrální server. To je vhodné pro velmi robustní a rozsáhlé sítě s velkým počtem zařízení. Dohledový systém je přístupný z webového rozhraní, které slouží zároveň i jako administrační prostředí pro správu a vyhodnocení dat. [6]

Zajímavý seriál zabývající se právě systémem Zabbix je publikován na webu společnosti Linuxsoft (url: [http://www.linuxsoft.cz/article.php?id\\_article=1963](http://www.linuxsoft.cz/article.php?id_article=1963)), kde je detailně probrána instalace, konfigurace a práce s tímto systémem. Část seriálu se dokonce zabývá i nároky na výpočetní výkon serveru v závislosti na počtu monitorovaných zařízení.

Dalším systémem stojícím za zmínění je Nagios, který umožňuje monitorovat počítačovou síť a v ní poskytované služby. V případě výskytu problému okamžitě informovat administrátora, který tak může rychle zasáhnout. Monitorovací služba periodicky spouští kontroly specifikovaných koncových uzlů a služeb. Používá k tomu externí moduly, které oznamují výsledek kontroly hlavnímu modulu Nagiosu. Pokud se vyskytne problém, služba pošle upozornění na předdefinované kontakty pomocí různých typů komunikace (email, SMS, nebo online zprávy – např. ICQ). Aktuální stav, historii záznamů a další výstupy jsou přístupné přes webové rozhraní. Umožňuje monitorování síťových služeb (SMTP, POP3, HTTP, NNTP, ICMP, SNMP), systémových prostředků (vytížení CPU, využitá kapacita pevného disku, logování systému) a umí monitorovat i operační systém Microsoft Windows s pluginem NRPE\_NT. Nabízí možnost vzdáleného monitorování přes protokol SSH nebo přes zašifrovaný SSL tunel. Pluginy jsou navrženy tak, aby si uživatelé mohli vytvářet vlastní funkcionalitu v různých jazycích (Bash, C++, Perl, Ruby, Python, PHP, C#). K upozornění o problémech a výpadech lze využít pageru, e-mailu, SMS nebo VoIP (napojení na systém Asterisk). V případě problému nabízí aktivní řešení pomocí Event handleru, například pro automatický restart služby. [7]

## 4.1 Freenetis

Freenetis je informační systém pro počítačové sítě neziskových organizací, jako jsou vnitřní sítě škol, internátů, VŠ kolejí komunitních sítí apod. Vývoj je pod svobodnou licencí GNU/GPL a je zveřejněn na portálech SourceForge a GitHub. Je vyvinut v jazyce PHP s využitím MVC architektury pomocí frameworku Kohana (verze 2.1) a pro správu dat se využívá MySQL databáze. Návrh a vývoj byl prováděn v rámci kvalifikačních prací na FAI UTB ve Zlíně a FEL ČVUT v Praze, v současné době je dále udržován a vyvíjen studenty FAI UTB ve Zlíně a FIT VUT v Brně. [1]

### 4.1.1 Zabezpečení a uživatelské účty

Freenetis v současné verzi neposkytuje pro nepřihlášené návštěvníky (hosty) téměř žádné funkce. Pro přístup k většině funkcí je proto nutné přihlášení. Systém disponuje kromě uživatelských účtů také skupinami, kterým jsou poté definovány práva a povinnosti. Jako povinnosti jsou myšleny členské příspěvky pro chod organizace.

Aplikace pro přístup nových členů umožňuje registraci. Po schválení registrace má nový uživatel zároveň založený i účet pro převody a správu peněz. Také je možné si na

každý takovýto účet registrovat další uživatele, kteří budou mít do aplikace přístup v rámci jednoho účtu.

#### 4.1.2 Zařízení a nástroje

Hlavním smyslem Freenetisu je správa síťových zařízení jako jsou počítače, routery, switche, huby, mobilní zařízení a přístupové body. Spravovat je možné také síťová rozhraní, IP adresy sítí a podsítí, porty zařízení, DHCP servery, VLAN rozhraní a sítě, bridge, ale i kabeláže. U zařízení je možné vzdáleně monitorovat jejich aktivitu, v případě výpadků upozornit správce a zobrazovat statistiky provozu. [1]

Kromě síťových zařízení je možné spravovat i mobilní nebo VoIP telefony a jejich služby včetně správy osobních a firemních hovorů. Stačí se pouze připojit na ústřednu poskytovatele a začít využívat jeho služeb.

#### 4.1.3 Instalační balíčky

System Freenetis je distribuován formou balíčků, které se instalují na platformě Debian operačního systému Linux. Základním balíkem je webová aplikace, která se může rozšířit o další nástroje. Za zmínku stojí zejména dva balíky Monitoring a Redirection, které by se mohly pro budoucí použití v projektu monitorování kotlů uplatnit.

##### 4.1.3.1 Monitoring

Balíček monitoringu slouží pro sledování dostupnosti jednotlivých zařízení v síti Freenetis a je spuštěn na serveru jako služba v pozadí. Zařízení se seskupují podle priorit a pro každou skupinu je vytvořen vlastní proces, který jednotlivá zařízení sleduje. Využívá nástroje *fping*, který vytváří požadavky na vzdálená zařízení pomocí IP adresy a získává dobu odezvy. Výsledky jsou zobrazeny v systémové administraci.

Monitorovací balíček může při velkém množství zařízení velmi vytěžovat server, na kterém je spuštěn. Autoři nedoporučují instalaci na stejný server, kde je spuštěn i Freenetis. Spojení mezi Freenetisem a tímto balíčkem je zajištěno pomocí funkce *wget*, takže při vzdálené komunikaci mezi těmito servery nevznikají problémy. [1]

Součástí balíčku je i nástroj odesílající automatické emaily se zprávami o výpadcích a opětovné dostupnosti zařízení, který posílá reporty v minutových intervalech.

## **II. PRAKTICKÁ ČÁST**

## 5 PŘENOS DAT Z KOTLE

Kotle pro vzdálenou komunikaci používají knihovnu „HTTP client“, která je součástí operačního systému Sigmatek LASAL. Během chodu programu se vyhodnocují data, která kotel přijímá od vnějších periférií, ale i od dalších kotlů. Při změně vnitřního stavu se vysílají události, jako je třeba chyba připojení (viz. kapitola 7.1). V takovém případě se přímo v kódu volá funkce „LogToHttp“ na objektu „cHTTP“. První parametr je identifikátor události konstantou a dalších pět parametrů je volitelných podle typu události.

```
cHTTP.LogToHttp(HTTP_EID_CASCADE_CLIENT_ONLINE,to_dint(ClientIndex),0,0,0,0);
```

Zpráva není ihned odeslána, ale je uložena do fronty. Pokus o odeslání zpráv probíhá každé dvě sekundy a pouze v případě, že má zařízení dostatek procesorového času pro odeslání požadavku a přijetí odpovědi. Toto řešení je aplikováno z důvodu vysokého vytížení procesoru právě při řešení paralelizace a komunikace mezi jednotlivými zařízeními. Z důvodu malého množství operační paměti je i omezena délka fronty na dvacet položek. Může tedy dojít k přetečení a ztrátě některých záznamů o událostech před jejich odesláním serveru.

```
FUNCTION GLOBAL HTTP_logging::LogToHttp
VAR_INPUT
    event_id : DINT;
    par1 : DINT;
    par2 : DINT;
    par3 : DINT;
    par4 : DINT;
    par5 : DINT;
END_VAR

VAR_OUTPUT
    Result : BOOL;
END_VAR

Result := false;
if(sLoggingEnabled <> 0) then
    cSysDate := cSysDate.Read(); //Načtení systémového času
    cSysTime := cSysTime.Read();
    if EventCounter < 1000 then EventCounter += 1; //Nastavení pořadí události
    else EventCounter := 0;
    end_if;
    if Q_Head = Q_Max then //Hledání ukazatele zásobníku
        if Q_Tail <> 0 then //omezení přepisování fronty při zaplnění
```

```

        Q_Head := 0;           //Nastavení ukazatele zásobníku
        Result := true;
    end_if;
    elsif Q_Tail <> (Q_Head + 1) then //omezení přepisování fronty
        Q_Head += 1;
        Result := true;
    end_if;
if Result == true then           //Naplnění zásobníku daty
    Q_HasItems := true;
    buffer[Q_Head][0] := cSysDate;
    buffer[Q_Head][1] := cSysTime;
    buffer[Q_Head][2] := sSystemID;
    buffer[Q_Head][3] := EventCounter;
    buffer[Q_Head][4] := event_id;
    buffer[Q_Head][5] := par1;
    buffer[Q_Head][6] := par2;
    buffer[Q_Head][7] := par3;
    buffer[Q_Head][8] := par4;
    buffer[Q_Head][9] := par5;
end_if;
END_FUNCTION

```

Operační systém volá v intervalu dvou minut funkci „Background“ na objektu „HTTP\_logging“ pro odeslání všech událostí, která vyhodnotí, jestli na zásobníku existují události, které je třeba odeslat. Zpráva se odesílá jedna po druhé, dokud je na procesoru dostatek volného času k dokončení operace.

```

FUNCTION VIRTUAL GLOBAL HTTP_logging::Background
VAR_INPUT
    EAX : UDINT;
END_VAR
VAR_OUTPUT
    state (EAX) : UDINT;
END_VAR

if (sLoggingEnabled <> 0) & Q_HasItems then
    if Q_Tail = Q_Max then
        Q_Tail := 0;
    else
        Q_Tail += 1;
    end_if;

    HTTPSend();

```

```

    if Q_Tail = Q_Head then
        Q_HasItems := false;
    end_if;
end_if;

state := READY;

END_FUNCTION

```

Následně se nachystají parametry v hexadecimálním formátu a vytvoří se spojení se serverem. Jelikož nedochází ke vzdálenému řízení ze serveru, tak se získaná odpověď jinak nezpracovává.

```

FUNCTION GLOBAL HTTP_logging::HTTPSend
VAR
    pHTTPCli      : ^OS_HTTP_CLIENT;
    retv          : SYS_ERROR;
    code         : DINT;
    rc           : DINT;
    sURL         : array[0..200] of char;
    sData        : array[0..100] of char;
    sNo          : ARRAY [0..20] of char;
    OutBuf       : array[0..10] of char;
    I            : DINT;

END_VAR

retv := OS_CILGET("HTTPCLIENT", #pHTTPCli$void);

if retv <> SYS_ERR_NONE then
    pHTTPCli := 0$^OS_HTTP_CLIENT;
else
    _strcpy(#sURL[0], "http://ponast.cz/boiler/receiver.php?data=");

    DtoH(#sNo[0],buffer[Q_Tail][0], 16#000, sizeof(char));
    _strcpy(#sData[0],#sNo[0]);
    for i:=1 to 9 do
        _strcat(#sData[0],"_");
        DtoH(#sNo[0],buffer[Q_Tail][i], 16#000, sizeof(char));
        _strcat(#sData[0],#sNo[0]);
    end_for;
    _strcat(#sURL[0],#sData[0]);
    rc := pHTTPCli^.HTTPGet $ G_HTTPGet(#sURL[0], #OutBuf[0], 10, #code);
end_if;
OS_CILRELEASE("HTTPCLIENT");

```

```
END_FUNCTION
```

Ještě před spuštěním kotle je potřeba inicializovat ukazatele do paměti sloužící pro ukládání zpráv do zásobníku a nastavit velikost zásobníku.

```
FUNCTION VIRTUAL GLOBAL HTTP_logging::Init
if _firstscan then
    EventCounter := 0;
    Q_Head := 0;
    Q_Tail := 0;
    Q_Max := 20;           //maximální počet zpráv v zásobníku
    Q_HasItems := false;
end_if;
END_FUNCTION
```

## 6 IMPLEMENTACE A DOSTUPNOST SYSTÉMU FREENETIS

System Freenetis je dostupný jak pro operační systém Windows, tak i pro Linux. Optimalizován je však pro distribuci Linuxového systému Debian.[2]

Poslední oficiální distribuce v současné době je verze *1.1.6* vydaná 19.3.2014, avšak pro implementaci byla použita doposud neoficiální verze *1.2.0~alpha1*, která má oproti oficiální verzi databázi založenou na MySQL funkci table partitioning. Předpokládá se, že každý kotel bude posílat parametry každých pět minut, což za rok činí 105 120 záznamů z každého zařízení. V případě desítek zařízení, se již pohybujeme v miliónech záznamů. Pokaždé, když by se program z této tabulky snažil číst, musela by MySQL nejprve všechny data načíst z disku do operační paměti, jelikož by byly uloženy jako jeden soubor. Pomocí rozdělení tabulky na oddíly se rapidně zmenší počet dat, které databáze musí načíst z disků a které nejsou použity, jelikož se otevrou pouze některé oddíly seřazené podle klíčového sloupce. Více informací lze nalézt přímo na stránkách MySQL na adrese: <https://dev.mysql.com/doc/refman/5.1/en/partitioning.html>. [3]

Instalace a vývoj byla prováděna na systému Windows 7, kde byl již přeinstalován program EasyPHP, což je balík s distribucemi PHP, MySQL a Apache 2 serveru. Ke stažení aplikace Freenetis je potřeba ještě nainstalovat program Subversion (SVN), který není součástí operačního systému Windows a umožňuje stahování nebo aktualizaci projektových souborů z repozitářů vývojářů. Následujícími příkazy v konzoli (příkazové řádce systému) se provede stažení aplikace a vytvoření potřebných adresářů pro její spuštění včetně nastavení přístupových práv. [2]

```
cd C:/www
svn checkout svn+ssh://svn.code.sf.net/p/freenetis/code/freenetis/branches/1.2
freenetis
cd www/freenetis
chmod ugo+w . upload
mkdir -m 0777 logs
```

Balíčky (viz. kapitola 4.1.3), které jsou aplikaci k dispozici, jsou optimalizovány pro operační systém Linux a využívají plánovače úloh CRON. Není možné je tedy instalovat na platformě Windows, avšak ani nebudou zapotřebí pro splnění stávající požadavků.

Při prvním spuštění se zobrazí instalační průvodce, kterým se během pěti kroků nastaví databáze, administrátorský účet, informaci o organizaci, poplatky a poté se provede samotná instalace.

**FreenetIS 1.2.0~alpha1**

**Instalace**

Vítejte v instalaci FreenetIS. Prosím, vyplňte formulář údajů o vašem sdružení

**Přihlašovací údaje**

**Přihl. jméno:**

**Heslo:**

**Heslo znovu:**

**Informace o sdružení**

**Název sdružení:**

**Datum založení:**

**Ulice:**

**ČP.:**

**Město:**

**Městská část:**

**PSČ:**

**Telefon:**

**E-mail:**

**Údaje o prvním bankovním účtu**

**Název účtu:**

**Číslo účtu:**

**Kód banky:**

**IBAN:**

**SWIFT:**

**Poplatky**

**Den stržení:**

**Vstupní příspěvek:**

**Měsíční členský příspěvek:**

**Transakční poplatek:**

**Pokuta:**

**Systém**

**Titulek stránky:**

**Země:**

**Měna:**

**Instalovat**

Obrázek 2 Webový instalační formulář aplikace Freenetis

Výhodou Freenetisu je i jednoduchá aktualizace za použití příkazu konzole.

```
cd C:/www/freenetis
svn update
```

Po aktualizaci zdrojových kódů se při následujícím spuštění aplikace automaticky aktualizuje databáze a nastavení systému.

## 7 KOMUNIKAČNÍ ROZHRAŇÍ

Kotle komunikují s rozhraním webové aplikace pomocí HTTP požadavků. Odesílání parametrů ze strany kotlů je již realizováno pomocí zasílání dat v proměnné GET (data jsou tak přenášena přímo v textovém řetězci umístěném jako součást URL adresy). Jelikož nebyla možnost se doposud se zařízením setkat, kvůli jeho neustálému vývoji, bylo třeba se přizpůsobit již existujícímu řešení přenosu dat, ale navrhnout aplikaci webu tak, aby mohlo dojít k inovaci a zvýšení množství přenášených dat.

### 7.1 Současná implementace

Všechna data jsou přenášena v jedné GET proměnné, nazvané „data“. Před zahájením přenosu jsou hodnoty jednotlivých parametrů převedeny do řetězce s hexadecimálním formátem, oddělené podtržítkem. Přenášených parametrů je deset a mají přesně dané pořadí, ve kterém jsou přenášeny. Prvních pět slouží pro identifikaci zařízení a události. Zbýlých pět parametrů jsou číselné hodnoty náležící k dané události. Identifikační parametry jsou:

1. **Systémové datum**
2. **Systémový čas**
3. **Identifikátor zařízení** - Unikátní ID pro spárování s webovou aplikací.
4. **Čítač události** - Některé události mohou být ztraceny z důvodu přetečení zásobníku na kotli důsledkem nedostatku procesorového času nebo paměti potřebných pro odesílání dat. Čítač událostí by měl zachytit toto přetečení. Je omezen na rozsah 0-1000. Nezabrání ztrátě, pouze ji indikuje.
5. **Typ události** – Události v systému jsou definovány jako číselné hodnoty, kterým je poté přiřazeno oznámení v textové podobě.

Výsledná sestavená URL adresa může vypadat následovně:

```
http://domena.cz/receiver.php?data=7DE30D4_6363A00_1_26_10_0_4_0_0_0
```

Události na straně zařízení jsou dány jako konstanty s jasným významem definované přímo v kódech a s možností dalšího rozšíření:

```
#define HTTP_EID_CASCADE_START 1
#define HTTP_EID_CASCADE_STABILIZATION_START 2
#define HTTP_EID_CASCADE_STABILIZATION_END 3
#define HTTP_EID_CASCADE_REINDEXATION 4
```

```
#define HTTP_EID_CASCADE_REINDEXATION_BY_TIMER 5
#define HTTP_EID_CASCADE_SYSTEM_START 6
#define HTTP_EID_CASCADE_SYSTEM_END 7
#define HTTP_EID_CASCADE_REQUEST_ON 8
#define HTTP_EID_CASCADE_REQUEST_OFF 9
#define HTTP_EID_CASCADE_REQ_TEMP_CHANGE 10
#define HTTP_EID_CASCADE_CHECK_REF_SYSTEM_CHANGE 11
#define HTTP_EID_CASCADE_CLIENT_ERROR_CONNECTION 12
#define HTTP_EID_CASCADE_CLIENT_MAX_ERRORS_CONNECTION 13
#define HTTP_EID_CASCADE_CLIENT_CASCADE_ERROR 14
#define HTTP_EID_CASCADE_CLIENT_ONLINE 15
#define HTTP_EID_CASCADE_CLIENT_STATUS_CHANGE 16
```

Ke každé události, podle jejího významu, jsou přiřazeny parametry, kterých může být až pět. Maximální počet parametrů je pevně daný, což může v budoucnu způsobit problémy při rozšiřování. Těmito parametry mohou být například při události spuštění identifikátory kotlů seřazené podle jejich priorit. Pokud by do průmyslové haly bylo namontováno více než pět kotlů, tak o dalších již nebude webová aplikace informována, jelikož jejich identifikátory se v logu nebudou přenášet.

## 7.2 Inovace současné implementace

Předchozí řešení vůbec nepočítá s možností logování parametrů v pravidelných intervalech, které nejsou nijak spjaté s událostmi zařízení. Mezi ně můžou patřit teploty, průtoky a stavy komponent kotlů. Pro takové parametry by bylo vhodnější navrhnout robustnější komunikační rozhraní, zejména pro předávání dat.

Další slabou stránkou může být kolize unikátních identifikátorů zařízení kotlů. Identifikátory jsou pevně ukotveny ve zdrojovém kódu. Je tedy nutné pro každou novou sestavu zařízení znovu kompilovat program s jiným unikátním identifikátorem.

System Sigmatek LASAL nabízí knihovny pro komunikaci přes TCP/IP. Aby zařízení mohlo komunikovat po TCP/IP, musí mít síťovou kartu. Avšak každá karta má vlastní světově unikátní MAC adresu (celkem  $2.8e^{14}$  možných kombinací adres). Pokud bychom zohlednili fakt, že některá zařízení umožňují změnu MAC adresy síťové karty, mohla by se ještě testovat i IP adresa (celkem  $4.2e^9$  možných kombinací adres) adresa sítě, ve které je MAC adresa umístěna. Tím dostaneme pravděpodobnost kolize  $1:1.2e^{24}$

Způsob předávání dat metodou GET v HTTP protokolu se může klidně zachovat a zároveň by se dalo využít možnosti předávání dat jako pole. Pokud přenášíme data

metodou GET nebo POST, tak se parametry převádí do jednoho textového řetězce, ale při zpracování na straně serveru se opět převedou zpět do pole. Přenášením dat pomocí polí můžeme zároveň odbourat nevýhodu omezení počtu parametrů. Data přenášovaná od zařízení do webové aplikace by tak mohla mít následující formát:

```
$_GET = array(  
    'ip' => '192.168.3.1',  
    'mac' => '12:34:56:78:90:ab',  
    'timestamp' => 1397302970,  
    'data' => array(  
        'params' => array(  
            'temp_out' => 32,  
            'temp_in_min' => 12,  
            'temp_in_max' => 28,  
            'status' => 'running'  
        ),  
        'logs' => array(  
            1=>array(  
                'number' => 1,  
                'type' => 1,  
                'params' => array(  
                    1 => 'value 1',  
                    2 => 'value 2',  
                )  
            ),  
            2=>array(  
                'number' => 2,  
                'type' => 7,  
                'params' => array(  
                    1 => 'value 1'  
                )  
            )  
        )  
    )  
);
```

Timestamp představuje časové razítko společné pro všechny přenášené data. Data jsou poté rozdělené na parametry a logy.

Parametry jsou nezávislé hodnoty ze systému, které se mohou posílat periodicky a díky tomu, že se jedná o asociativní pole, tak nezáleží na jejich počtu. Klíče asociativního pole poté slouží k překladu názvu na uživateli srozumitelný název s popisem.

Logy představují záznamy o událostech, kde se vychází z již realizovaného řešení. Každý log představuje několik parametrů, které jsou vnořeny do dalšího pole. První pole slouží pouze pro umožnění zaslání několika logů ve stejný časový okamžik, kde klíč představuje pořadové číslo a hodnota je jako výčet vlastností logované události.

- **Number** – pořadové číslo logované události, kvůli zpětné kompatibilitě a zachycení informace o ztrátě některých logů v zařízení, způsobených přetečením zásobníku
- **Type** – číselná hodnota značící typ události, které jsou ve výčtu v kapitole 7.1
- **Params** – seznam parametrů, které se k dané události přenášejí. Ruší se tím omezení na počet přenášených parametrů

### 7.3 Příjem dat

Na straně webové aplikace je vytvořený jednoduchý skript psaný v jazyce PHP, který automaticky přijme data posílané v HTTP požadavku, zpracuje je, ověří zařízení odesílající data a pokud proběhne ověření v pořádku, tak data k příslušnému zařízení uloží. Aplikace se spouští za pomoci frameworku Kohana, nicméně tento skript framework nevyužívá, aby se maximalizovala rychlost zpracování požadavků. Při testování na lokálním serveru bez akceleratorů PHP se dosáhlo rychlosti zpracování 3076 požadavků za minutu, kde doba odezvy na zpracování jednoho požadavku činila 19,5 milisekund. Při měření bylo odesláno celkem 17 000 požadavků. Na produkčním serveru bude tato rychlost 2-3x vyšší. Pokud by ale v budoucnu tato propustnost byla malá, bylo by lepší skript napsat ve formátu fastCGI, který umožní dosáhnout podstatně lepších výsledků, popřípadě využít mezipaměti a následného dávkového zpracování požadavků.

#### 7.3.1 Skript pro zpracování

Je realizován jeden skript pro zpracování dat, který je v rámci aplikace umístěn na URL adrese:

```
http://nazev-domeny.cz/boiler/receiver.php
```

Algoritmus rozpozná, jestli data jsou posílány současným způsobem nebo inovovaným a automaticky data zpracuje a uloží.

### 7.3.2 Autorizace

Před samotným uložením dat se provede autorizace vysílajícího zařízení. Systém Freenetis nabízí možnost identifikace zařízení pomocí MAC adres ve formátu „12:34:56:78:90:ab“. Současná implementace komunikace nenabízí možnost vysílání MAC ani IP adres, ale jen unikátního identifikátoru v číselné podobě. Ten se proto musí převádět z číselného formátu například „16482“ na „00:00:00:01:64:82“ a takto převedený formát identifikátoru na MAC adresu je nutné zadat ve webovém rozhraní při registraci nového zařízení, jinak by autorizace neproběhla v pořádku.

### 7.3.3 Identifikace nových typů událostí

Postupným vývojem software zařízení kotlů na tuhá paliva mohou přibývat nové parametry a události, které bude potřeba zavést i do systému webové aplikace. Z důvodu zachování konzistence dat v databázi se při ukládání nové události ověří, zda daný typ události v systému existuje. Pokud ne, tak se vytvoří nový záznam, ve kterém je vyplněn pouze identifikátor události a administrátoři poté doplní detaily, které se z důvodu urychlení přenosu dat a minimalizaci nároků nepřenášejí.

## 8 VLASTNÍ ŘEŠENÍ APLIKACE

Projekt Freenetis je rozsáhlý, ale i přesto bylo potřeba provést rozšíření současné databáze a MVC architektury. V následujících kapitolách budou popsány závislosti mezi tabulkami, objekty a jejich realizace v databázové i aplikační vrstvě. Změny byly provedeny i v souborovém systému. Několik souborů přibylo a jiné musely být upraveny. Seznam změn uvedený níže je vztážený vůči kořenovému adresáři aplikace Freenetis.

### Nové soubory a adresáře:

- */application/controllers/device\_boiler.php* – kontrolér řídící zpracování dat v modelech před vykreslením administračních stránek
- */application/models/device\_boiler\_graph.php* – předzpracování dat pro vykreslení vybraných parametrů do grafů
- */application/models/device\_boiler\_log.php* – model spravující logované parametry ke každému kotli
- */application/models/device\_boiler\_message.php* – správa událostí z kotlů
- */application/models/device\_boiler\_message\_type.php* – zpracování a vedení zpráv o událostech
- */application/models/device\_boiler\_param.php* – správa seznamu logovaných parametrů
- */application/views/device\_boiler/graph.php* – vykreslení a vizualizace grafů
- */application/views/device\_boiler/messages.php* – vykreslení výpisů oznámení
- */application/views/device\_boiler/messages\_wrapper.php* – vykreslení detailů zpráv ze zařízení
- */application/views/device\_boiler/param\_edit.php* – vykreslení formuláře pro nastavení parametrů kotlů
- */boiler/.htaccess* - řízení a povolení přístupu do adresáře na Apache serveru
- */boiler/Boiler.php* - třída ověřující a zpracovávající přijímané data z kotle
- */boiler/generator.php* - generátor náhodných dat pro testovací účely
- */boiler/receiver.php* - rozhraní pro příjem a ověření dat z kotle

### Změněné soubory:

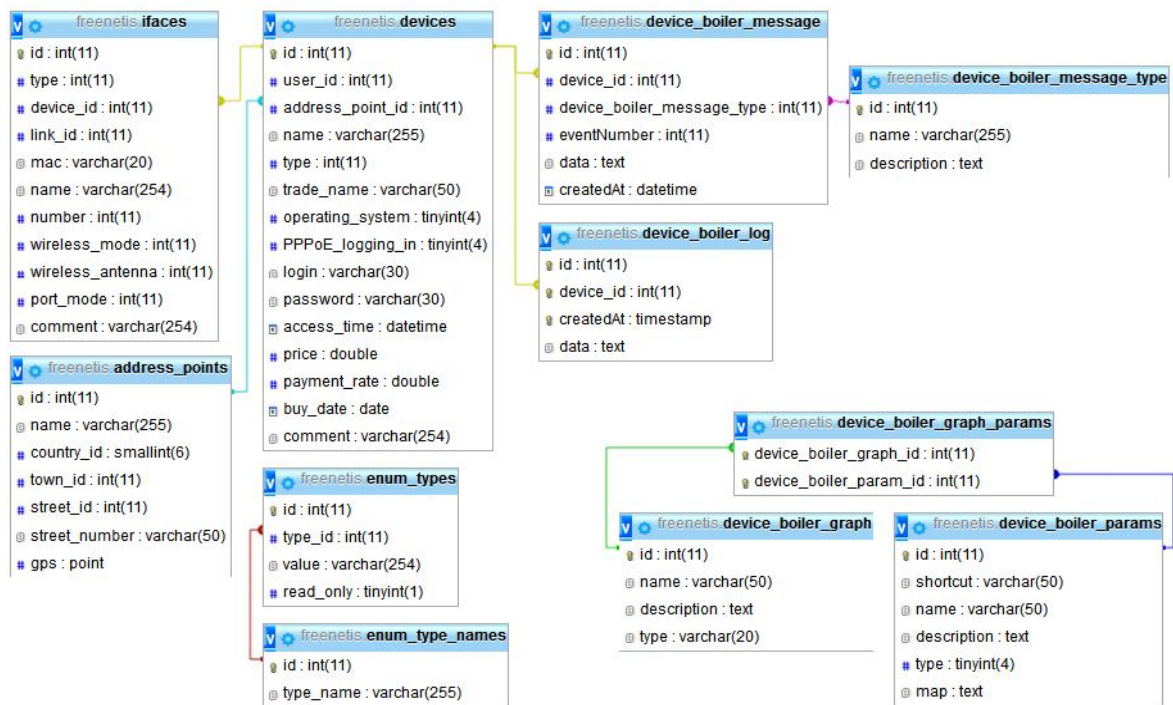
- */application/libraries/MY\_Controller.php* – funkce „build\_menu“ byla rozšířena o odkaz na výpis nejnovějších událostí

- `/application/views/device/show.php` – přidán odkaz na stránku s logovanými daty z kotlů

*Poznámka: Soubory umístěné v adresáři `/application/models` se starají i o přístup k databázovým tabulkám.*

## 8.1 Databázová struktura a aplikační logika

Databázová vrstva aplikace obsahuje 104 tabulek, jejichž strukturu by nemělo smysl zde uvádět. Všechny tabulky jsou typu InnoDB, díky čemuž umožňují vytvářet relace mezi tabulkami a v případě ORM v rámci celého projektu Freenetis tak vytvářet relace mezi objekty skrze databázi. Na následujícím obrázku jsou zobrazeny nejdůležitější tabulky a hlavní relace mezi nimi, které jsou realizovány na databázové vrstvě.



Obrázek 3 Vybrané tabulky a jejich relace na databázové vrstvě

Na levé polovině jsou tabulky systému Freenetis, které je nutné spravovat pro realizaci vlastního rozšíření, nebo je třeba znát jejich význam. Vpravo jsou tabulky pro zavedení monitorování zařízení. U všech přidávaných tabulek jsou nadefinovány cizí klíče s nastaveným kaskádovým mazáním. Pokud se tedy smaže některé ze zařízení, databáze automaticky smaže i záznamy z logů zpráv a parametrů, čímž se zamezí vzniku mrtvých nebo neidentifikovatelných záznamů.

### 8.1.1 Tabulka zařízení

Obsahuje záznamy o všech konkrétních zařízeních definovaných v systému (routery, switche, mobilní telefony, počítače, AP, atd.), a které jsou ještě rozděleny do samostatných kategorií. Rozlišování konkrétního druhu zařízení se provádí pomocí tabulky „enum\_types“, kde jsou jednotlivé druhy zařízení ještě seskupovány do kategorií. Mezi tabulkou „devices“ a „enum\_types“ není na databázové vrstvě žádná relace, jelikož je vytvářena až na konkrétních místech v programu aplikace.

U každého zařízení jsou přiřazeny informace o jejich ceně, splátkách, přihlašovací údaje, druh operačního systému, adresa a vlastník. Dalšími externími tabulkami jde každému zařízení přiřadit administrátora, který kontroluje jeho provoz a stav. Pro alokaci se využívá tabulky „address\_point“, ke které jsou definovány názvy měst, ulic a států, číslo popisné a samotné GPS souřadnice. Získávání GPS souřadnic podle adresy a obráceně se provádí pomocí služby map společnosti Google.

Každé síťové zařízení má rozhraní, což je port, do kterého můžeme připojit kabelem jiné zařízení v rámci určité sítě. Routery a switche takových portů mohou mít více. Všechny se definují v tabulce „ifaces“, kde je uvedena i MAC adresa. Přes tuto tabulku je možné dohledat IP adresy zařízení a sítě, ve kterých jsou zapojeny. Tyto údaje se používají pro ověřování a spárování zařízení před přijímáním dat.

Pro práci se zařízeními v systému se využívá i dalších tabulek a závislostí, které jsou ale pro realizaci rozšíření monitorování irelevantní jako jsou šablony zařízení pro každou jejich kategorii, administrace, zabezpečení, rychlostní třídy a další.

### 8.1.2 Události a parametry

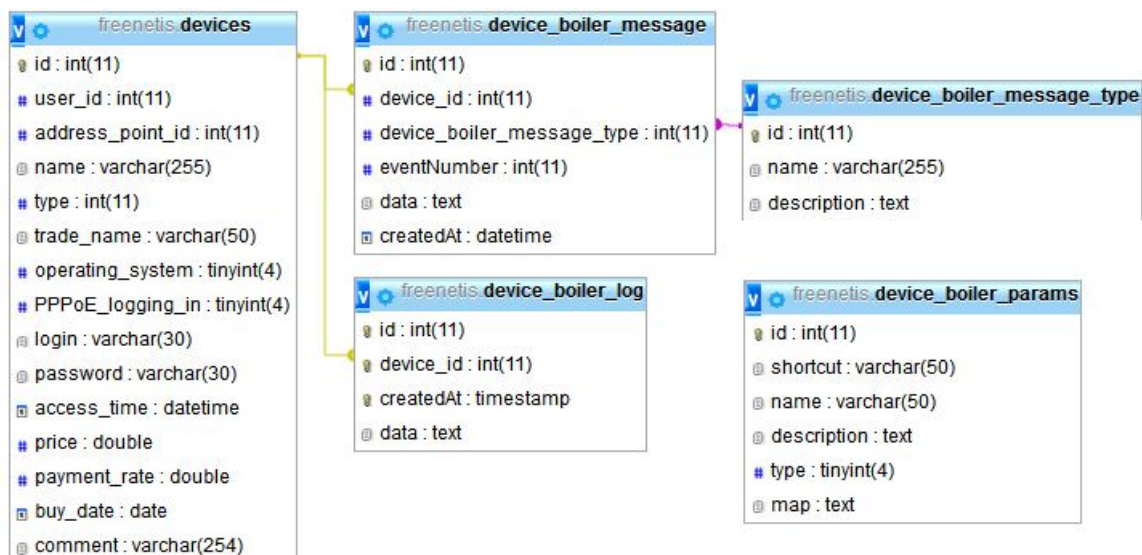
Zařízení spárované s aplikací může přes přijímací rozhraní předávat zprávy o svém stavu nebo události. Ty jsou spárovány za pomoci MAC adresy pomocí tabulky „ifaces“ a IP adresy tabulky „ip\_adresses“ se zařízeními v tabulce „devices“.

Události se poté přiřazují do tabulky „device\_boiler\_message“. Druhů zpráv o události může být několik desítek až stovek. Z důvodu minimalizace databáze a zásad návrhů databáze se definují pomocí číselné hodnoty, jejichž název a význam je definován v tabulce „device\_boiler\_message\_type“. Každá zpráva o události obsahuje informaci z čítače událostí v zařízení, čase vzniku a data, která jsou přiložena k události. Data jsou

uloženy ve formátu textu po převodu funkcí serialize jazyka PHP, jelikož nemusí být znám jejich rozsah (viz kapitola 8.1.2.1).

Pro záznam parametrů ze zařízení slouží tabulka „device\_boiler\_log“, kde se ukládá pouze záznam časového razítka a data serializované do textu.

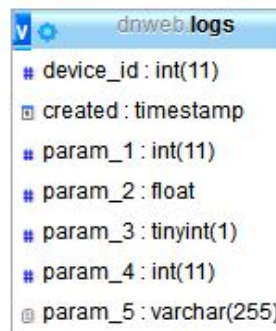
Poslední tabulkou této problematiky je „device\_boiler\_params“. Mezi ní a sloupci „data“ u tabulek „device\_boiler\_message“ a „device\_boiler\_log“ je v aplikační vrstvě vytvořená závislost. Při zobrazení dat se hodnoty ve sloupci „data“ (ve formátu asociativního pole „,klíč“ => ,hodnota““) pomocí klíče přiřadí záznamy z tabulky „device\_boiler\_params“, kde je klíč definován ve sloupci „shortcut“. Tato problematika je více vysvětlena v kapitole 8.1.3.



Obrázek 4 Závislosti tabulek pro uchování logů parametrů a událostí

### 8.1.2.1 Využití serializace a deserializace

Při návrhu databáze pro ukládání parametrů se nejčastěji využívá fixního návrhu databáze, kdy se vytvoří tabulka s daným počtem parametrů a jejich datovými typy, jako je na příkladu (Obrázek 5). Výhoda takového řešení je v přehlednosti, jednodušším hledání v databázi. Problém nastane, pokud potřebujeme například pro jeden případ logu přidat ještě dalších pět parametrů. Jsme pak nuceni přepracovat veškerou aplikační logiku, která pracuje s logy. Další, na co musíme pamatovat, je zachování pořadí všech parametrů. To může být problém zejména při verzování programu kotlů.



Obrázek 5 Ukázka tabulky s fixními parametry

Databázovou tabulku si například v jazyce PHP můžeme interpretovat jako dvě vnořená pole, kde první shromažďuje řádky a druhé sloupce.

```
$table = array(
    0=>array('device_id','created','param_1','param_2','param_3','param_4'),
    1=>array(1,'2014-03-02 13:25:35', 7,5,15,37),
    2=>array(1,'2014-03-02 13:30:02', 8,5,16,40)
);
```

V takovém případě můžeme sloupce parametrů vytáhnout do dalšího samostatného pole a to ukládat jako asociativní pole.

```
$table = array(
    0=>array('device_id','created','data'),
    1=>array(1,'2014-03-02 13:25:35', array(
        'param_1'=>7,
        'param_2'=>5,
        'param_3'=>15,
        'param_4'=>37)
    ),
    2=>array(1,'2014-03-02 13:30:02', array(
        'param_1'=>8,
        'param_2'=>5,
        'param_3'=>16,
        'param_4'=>40)
    )
);
```

Data uložená v takovém poli není možné přímo vkládat do databáze, jelikož MySQL neumí pracovat s poli. Data se musí převést na textový formát funkcí `serialize` a při čtení zpět funkcí `deserialize`. Tyto funkce umožní převod čísel, řetězců, polí ale i objektů včetně některých jejich závislostí do textu a zpět. Jsou přitom zachovány datové

typy a nastavení těchto struktur. Pro příklad je zde uvedena část programu, kterou si tuto funkčnost můžeme ověřit.

```
$data = array(
    'param_1'=>8,
    'param_2'=>5,
    'param_3'=>16,
    'param_4'=>40
);
$serializedData = serialize($data);
echo $serializedData."<br><br>";
if(unserialize($serializedData) === $data)echo "Data nebyla změněna";
```

Výsledkem takového kódu je:

```
a:4:{s:7:"param_1";i:8;s:7:"param_2";i:5;s:7:"param_3";i:16;s:7:"param_4";i:40;}
Data nebyla změněna
```

Návrh tímto řešením má ale dvě podstatné nevýhody. První je nárůst velikosti logu s parametry, jelikož s každou hodnotou se ukládá i její klíč. Nezáleží však na pořadí dat při jejich zpracování. Druhou nevýhodou může být hledání parametrů například pro jejich součet. K vykonání takovéto operace je potřeba načíst všechny řádky z tabulky v zadaném rozsahu, každý datový sloupec deserializovat a s hodnotami provést požadovanou operaci. V případě velkého množství dat tak může dojít k velkým latencím. Operace se nedá provést v databázové vrstvě, jelikož MySQL nezná serializaci.

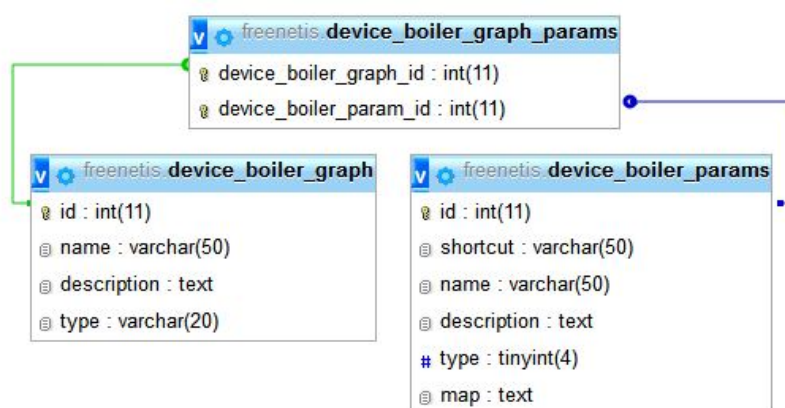
### 8.1.3 Grafy

Zaznamenané serializované parametry je potřeba zobrazovat do grafů, aby se daly sledovat trendy průběhu hodnot. Grafy vznikají jako záznamy v tabulce „device\_boiler\_graph“. Stačí definice názvů, popisu grafu a typu grafu. Typ grafu je řetězec, který přejímá model vykreslující data.

Parametry grafu musí být definovány zvlášť. K tomuto slouží tabulka „device\_boiler\_param“. Pro identifikaci parametru se využívá textového sloupce „shortcut“, který slouží pro mapování serializovaných dat, které jsou součástí zpráv o událostech a v logovaných parametrech. Sloupce „name“ a „description“ slouží jako vysvětlivky k parametrům, jelikož uložené zkratky mohou mít pro nezasvěceného uživatele nesmyslný význam. Sloupec „type“ slouží k rozlišení datového typu parametru, zda se jedná o číselné nebo stavové hodnoty (viz kapitoly 2.1.1 a 2.1.2). U stavových

hodnot se využívá ještě sloupce „map“, kde jsou uloženy překlady hodnot získaných ze zařízení, na hodnoty srozumitelné uživateli (např: převod zkratky „CND“ na celý název „Connected new device“). To je zavedeno zejména kvůli úsporám operační paměti u kotlů. Tyto překlady mohou být realizovány pro zkratky ve formě řetězců, ale větší využití se předpokládá pro náhrady textových popisů za číselné indexy a zpět (např: 1=“Start“, 2=“Running“, 3=“Waining“, 4=“Stopped“, apod.)

V každém grafu je potřeba zobrazit 1 až N parametrů, například kvůli jejich porovnávání. K tomu je určena tabulka „device\_boiler\_graph\_params“, kde se pouze definují vazby mezi grafy a parametry. Jeden parametr tak může být zobrazen ve více grafech. Aplikační logika se poté stará o to, aby podle typu grafu byly vybírány jen některé parametry, které je možné korektně zpracovat a zobrazit. Například v koláčovém grafu nemá smysl zobrazovat data, která mohou nabývat hodnot reálných čísel.



Obrázek 6 Tabulky parametrů, grafů a jejich závislostí

#### 8.1.4 Rozšíření výchozích struktur

Pro zprovoznění nových funkcionalit se musely rozšířit již existující záznamy v databázi. Změny se provedly v následujících tabulkách:

- **translations** (tabulka překladů) – byly přidány překlady slov, používané v rozšíření aplikace
- **enum\_types** (kategorizace dat a objektů) – pro rozšíření o kategorii zařízení náležící kotlům přibyl následující záznam, který je spojen s kategorií „Device type“ v tabulce „enum\_type\_name“

id	type_id	value	read_only
82	2	Boiler	1

Tabulka 1 Rozšiřující záznamy databázové tabulky enum\_types

- **axo** (třídy a výsady pro zabezpečení) – do zabezpečení byly přidány nové třídy a výsady, na kterých se ověřuje oprávnění uživatele. Rozšíření umožní zabezpečit nově přidanou aplikační logiku

id	Section_value	value	name
256	Device_Boiler_Controller	logs	Logs
257	Device_Boiler_Controller	messages	Messages
258	Device_Boiler_Controller	params	Assign boiler parameters

Tabulka 2 Rozšíření databázové tabulky axo o nové záznamy

## 8.2 Uživatelské rozhraní

Aplikace obsahuje výchozí rozložení tak, jak je definováno po instalaci Freenetisu. V Hlavním menu v sekci administrace přibyl odkaz na stránku „Boilers“. Na této stránce je výpis posledních zpráv o událostech ze všech zařízení seřazené podle data vzniku sestupně a vždy jen poslední zpráva od každého zařízení. Zpráva obsahuje pouze datum a čas, název zařízení, které je i odkazem na detaily, a název události. Tato stránka slouží jako rozcestník pro administrátory, kteří mohou nahlížet na nejnovější události a zároveň jsou zde umístěny další odkazy na stránky pro nastavení událostí zařízení, parametrů a grafů.

Další rozšíření uživatelského rozhraní je vidět u prohlížení zařízení. Pokud v hlavním menu a záložce „Sít“ přejdeme na stránku „Zařízení“, zobrazí se seznam již existujících zařízení, které uživatel vlastní nebo je jejich správcem. Kromě názvu je zobrazen i typ zařízení. Při kliknutí na akci „zobrazení“ u zařízení typu „boiler“ se otevře stránka s detaily zařízení, na které je vidět nastavení, síťové umístění, rozhraní a seznam správců. Nad tabulkou nastavení přibyl v seznamu odkaz „Logs from device“, kterým se přesměrujeme na seznam událostí ze zařízení.

Členové » ID 1 - Petr Štípek » Uživatelé » pipaslot pipaslot (pipaslot) » Zařízení » Zlín Kotel

**Zařízení Zlín Kotel**Upravit | Smazat | Upozornění | Exportovat | Zobrazit strom podzařízení | Zobrazit topologii | **Logs from device**

ID zařízení	1
Člen	Petr Štípek (1)
Uživatel	pipaslot pipaslot
Výrobní název	KP 11.1
Typ	Boiler
Přihl. jméno	
Heslo	
Datum koupě	2014-03-06
Adresa umístění	Budovatelská 4816, Zlín, 76005, Czech Republic
GPS souřadnice	49°14'2.904"N, 17°40'5.333"E
Komentář	
Aktivní odkazy zařízení	



Obrázek 7 Náhled na detaily o zařízení typu kotel

V následujících odstavcích bude vysvětleno, jakým způsobem krok po kroku se systém nastavuje pro využití všech rozšíření systému Freenetis k monitorování kotlů.


**8.2.1 Přidání nového zařízení**


Všechny zařízení jsou kategorizovány podle typu (kotel, router, switch, atd.) a dále jsou třízena podle šablon. Šablona představuje konkrétní typ zařízení, které má vstupní a výstupní porty. Než se začne systém plnit jednotlivými zařízeními pro monitorování, je nejprve potřeba zaregistrovat jednotlivé šablony.

V menu „Administrace / šablony zařízení“ nalezneme seznam všech definovaných šablon podle typu zařízení. Kliknutím na „Přidat šablonu“ se zobrazí formulář (Obrázek 8), kde výrobní název je názvem série výrobků (například u firmy Ponast to jsou KP 11, KP 21, KP 51, atd.) a typ je kotel (boiler). Pokud vybereme položku „Výchozí pro tento typ zařízení“, tak se tato šablona automaticky nastaví pokaždé, když budeme registrovat nová zařízení typu kotel. Dalšími parametry jsou ethernetové rozhraní, bezdrátové rozhraní a porty. Ty jsou v závislosti na síťové kartě zařízení, ale u kotle to bude vždy jen jeden ethernetový port. Aby bylo možné realizovat síťové propojení v aplikaci, je potřeba zaregistrovat i šablony routerů a switchů, ke kterým se kotel následně připojí.

## Přidat novou šablonu zařízení

**Základní informace**

**Výrobní název:**  

**Typ:**  

Výchozí pro tento typ zařízení?

**Aktivní odkazy zařízení:**

---

**Ethernetová rozhraní**

**Počet:**

**Jméno:**

---

**Bezdrátová rozhraní**

**Minimální počet:**


**Maximální počet:**

---

**Porty**

**Počet:**

---

**Vnitřní rozhraní** 

---

Obrázek 8 Formulář pro vložení nové šablony zařízení

Nyní je potřeba si nachystat síť, ve kterých zařízení budou umístěny. V menu „Administrace / Podsítě“ je jejich seznam. K zadání sítě postačí znát IP adresu a masku, která určuje počet zařízení v síti. V rámci podsítě lze nastavit, zda je v ní umístěn DNS server nebo DHCP server. Tyto informace slouží pro zjednodušení hledání potíží při provozu sítě a není nutné je vyplňovat. Je nutné nastavit vlastníka, kterému síť patří, a případně můžeme přidat oblast, kde se zařízení nachází. Oblasti slouží jako kategorizace sítí v rámci geografického rozložení. Oblastí může být například městská část. Po uložení sítě se automaticky vygenerují IP adresy, na které se následně budou připojovat jednotlivá zařízení.

V tuto chvíli přichází na řadu registrace všech konkrétních zařízení, jejichž seznam je v menu „Administrace / Zařízení“. Nejprve zaregistrujte routery a switche, ke kterým má být připojen některý z kotlů. Název zařízení a vlastníci uživatel je volitelný. Vyberte typ a šablonu. U routeru je možné vyplnit přihlašovací jméno a heslo do routeru a

nadefinovat správce zařízení. Pokud jste vybrali vlastního uživatele, tak se automaticky doplní adresa zařízení a GPS souřadnice. V opačném případě je vhodné je vyplnit jako informaci pro technika. Po potvrzení formuláře je možné nastavit rozhraní. Vyberte síť, ve které je umístěn, a zadejte jeho IP adresu, která náleží do některé z již vytvořených sítí.

### Přidat nové celé zařízení

**Zařízení**

<b>Název zařízení:</b>	<input type="text" value="router"/>
<b>Uživatel:</b>	----- Vyber uživatele -----
<b>Typ zařízení:</b>	router
<b>Šablona zařízení:</b>	Belkin N+ Wireless Router

*Detail zařízení* +

*Splátky zařízení* +

---

**Adresa**












<b>Město:</b>	----- Vyber město -----
<b>Ulice:</b>	<input type="text"/>
<b>ČP.:</b>	<input type="text"/>
<b>Země:</b>	Czech Republic
<b>GPS souřadnice X:</b>	<input type="text"/>
<b>GPS souřadnice Y:</b>	<input type="text"/>

Obrázek 9 Formulář pro přidání nového zařízení (krok 1)

Po uložení se ve formuláři zobrazí nová políčka pro nastavení rozhraní. Vždy jedno ethernetové rozhraní musí být připojeno k některé z existujících sítí. Zadaná IP adresa musí být z rozsahu vybrané sítě.

Stejným způsobem se vytváří zařízení kotlů. Je pouze nutné v druhém kroku vybrat zařízení, ke kterému je kotel připojen a vyplnit jeho MAC adresu. Identifikátor kotle je nutné přepsat do formátu MAC adresy. Například identifikátor „123456“ je nutné zapsat jako „00:00:00:12:34:56“, jinak systém nebude schopen rozeznat zařízení, které odesílá data a ty budou ztraceny.

## Přidat nové celé zařízení

Zařízení		
<b>Název zařízení:</b>	Boiler 	
<b>Uživatel:</b>	Petr Štípek	
<b>Typ zařízení:</b>	Boiler	
<b>Šablona zařízení:</b>	KP 11	
<b>Detail zařízení</b> 		
<b>Splátky zařízení</b> 		
<b>Cena:</b>	<input type="text"/>	
<b>Měsíční splátka:</b>	<input type="text"/>	
<b>Datum koupě:</b>	2014-05-07	
<b>Adresa</b>		
<b>Město:</b>	Rožnov pod Radhoštěm, 75661	
<b>Ulice:</b>	Kulturní	
<b>ČP.:</b>	1752	
<b>Země:</b>	Czech Republic	
<b>GPS souřadnice X:</b> 	49°27'56.549"	
<b>GPS souřadnice Y:</b> 	18°8'5.521"	
<b>Ethernetová rozhraní</b> 		
<b>Rozhraní</b>	<b>IP adresa</b>	<b>Připojeno k zařízení</b>
Jméno: eth1 	IP: <input type="text"/> 	---- Vyber zařízení ---- 
MAC: <input type="text"/>	Podsít: 192.168.1.0/27: Petr Štípek - Bud 	<input type="text"/> 

Obrázek 10 Formulář pro přidání nového zařízení (krok 2)

## 8.2.2 Nastavení přístupových práv

V systému jsou v tuto chvíli nadefinovány zařízení a síť. Je nutné nastavit práva skupinám uživatelů, kteří budou mít oprávnění zařízení zobrazovat a spravovat (například technici). Všechny stránky rozšíření jsou zabezpečeny stejně jako systém Freenetis. Nastavení oprávnění nalezneme v menu „Administrace / Přístupová práva“. Práva se nastavují definováním oprávnění (zobrazit, vytvořit, upravit, smazat), skupinami uživatel, pro které oprávnění budu platit a výběrem objektů zabezpečení (objekt kotlů: Device\_Boiler\_Controller).

Právo pro zobrazení výpisu ze zařízení je závislé na právu zobrazit zařízení. Automaticky tak k logům mají přístup majitelé zařízení, jejich správci a administrátoři, kteří mají přehled o všech zařízeních v systému.

### 8.2.3 Seznam událostí

Po spárování zařízení se systémem se automaticky začnou přijímat všechny parametry a zprávy o událostech. Pokud do systému dorazí nový druh události, automaticky se jeho identifikátor uloží a vytvoří se prázdný záznam. V menu „Administrace / Boilers / Logs“ je seznam všech typů událostí. Je možné přidat nové události před jejich prvním přijetím, nebo upravit doposud nenastavené události. Každá událost má číselný identifikátor, který musí být unikátní (systém si jej sám ověří), název, který se bude zobrazovat místo identifikátoru a popis.

### 8.2.4 Seznam parametrů

Podobně jako seznam událostí, i seznam parametrů slouží jako překlady přijímaných názvů parametrů na uživateli srozumitelný text. Unikátním identifikátorem není číslo, ale zkratka jako textový řetězec. Parametr je možno nastavit jako stavový, kdy se hodnota překládá na definovaný název. Pořadí zadávaných stavů je vždy nejprve původní hodnota přijatá od kotle a její překlad. Ve výchozím stavu je možné zadat jen tři stavové hodnoty, po uložení záznamu se zobrazí další tři formulářová pole pro zadání dalších překladů stavů.

**Parameter status**

shortcut	<input type="text" value="status"/>
Jméno	<input type="text" value="Stav"/>
Popis	<input type="text" value="Informace o běhu zařízení"/>
Stav	<input checked="" type="checkbox"/>
Stav 1	<input type="text" value="0"/> => <input type="text" value="Vypnuto"/>
Stav 2	<input type="text" value="1"/> => <input type="text" value="Spuštěno"/>
Stav 3	<input type="text" value="wait"/> => <input type="text" value="Čeká"/>
Stav 4	<input type="text"/> => <input type="text"/>
Stav 5	<input type="text"/> => <input type="text"/>
Stav 6	<input type="text"/> => <input type="text"/>

Obrázek 11 Definice nového stavového parametru

Pokud jsou už od zařízení přijaté některé hodnoty parametrů, ale nejsou stále v tabulce definovány, je možné některé dohledat. Přejdem na odkaz „Find new

parameters“, se spustí rutina, která prohledá několik poslední záznamů přijatých z kotlů, porovná parametry se seznamem a doplní nové záznamy.

## Boiler params

[Graphs](#) | [Find new parameters](#) | [Create new parameter](#)

Celkem položek: 7

Shortcut	Jméno	Popis	Stav	Akce
event_param_2	Kotel 2	ID kotle s pořadovým číslem 2	ne	 
event_param_1	Kotel 1	ID připojeného kotle	ne	 
status	status		ano	 
temp_with_error	temp_with_error		ne	 
temp_max	temp_max		ne	 
temp_min	Minimální teplota		ne	 
temp_out	Venkovní teplota	Z teploměru před vratnicí	ne	 

« předchozí **1** další »

50 ▼

Obrázek 12 Výstřižek stránky s výpisem parametrů

### 8.2.5 Nastavení a zobrazování grafů

Pro vykreslování charakteristik pomocí grafů je nutné mít nadefinované potřebné parametry. Každý graf se sestává z názvu, popisu, definice typu a samotných parametrů. Je možné zobrazit data ve třech typech grafů:

1. **Pie chart** – Koláčový graf určený pro zobrazení stavových parametrů, kde hodnoty jsou vykresleny jako procentuální vyjádření z vybraného období.
2. **Line chart** – Graf zobrazující hodnoty parametrů jako čáry se spojitým průběhem a časovou horizontální osou.
3. **Area chart** – Stejný jako Line chart, avšak jsou vyznačené ještě oblasti mezi hodnotami parametrů.

Každý graf může mít libovolný počet parametrů, avšak při ukládání změn je možné přidávat vždy jen po třech a poté je nutné uložení.

Nastavené grafy je možné zobrazit pro všechny šablony zařízení definovaných pod typem kotel (boiler).

## Graph: Status

<b>Jméno:</b>	Status
<b>Popis:</b>	Stav zařízení zobrazený jako procentuální poměr
<b>Typ:</b>	pie chart
<b>Parameter 1:</b>	status
<b>Add parameter:</b>	---
<b>Add parameter:</b>	---
<b>Add parameter:</b>	---

Odeslat

status

--- Vyber ---

**Hodnota**

Kotel 1

Kotel 2

Minimální teplota

temp\_max

temp\_with\_error

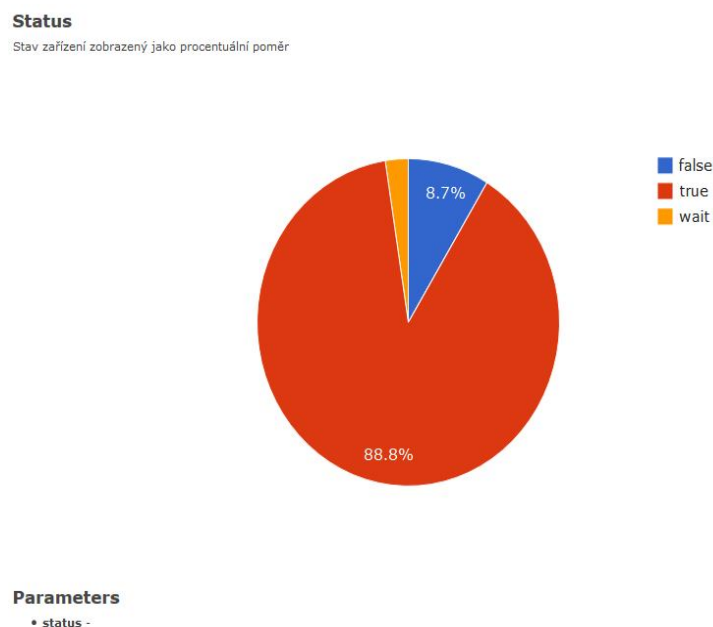
Venkovní teplota

**Stav**

status

Obrázek 13 Nastavení zobrazení grafů pro stavové parametry

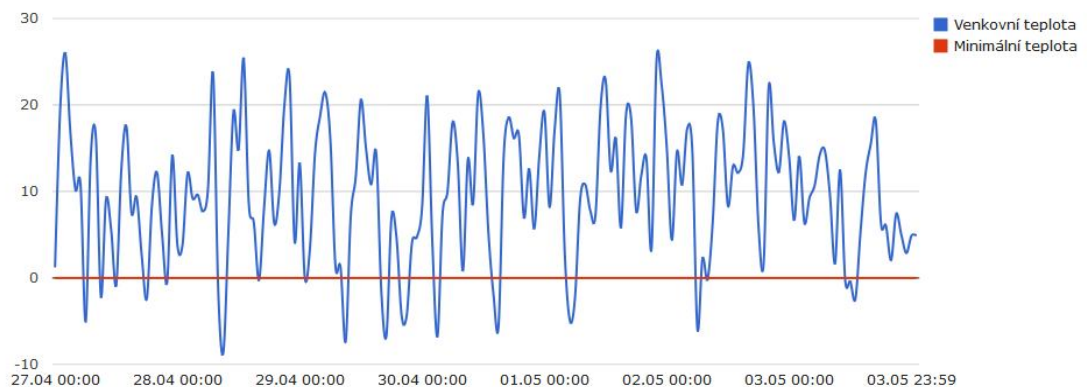
Následující tři obrázky (Obrázek 14, Obrázek 15, Obrázek 16) ukazují náhledy grafů pro náhodná data, včetně překladů parametrů a informací zadaných v jejich seznamech.



Obrázek 14 Ukázka koláčového grafu stavu zařízení pro jeden stavový parametr

### Teplota mimo halu

Monitorování venkovní teploty a vyhodnocování stavu vůči minimální a maximální hodnotě, která je povolena v danou dobu venku



### Parameters

- Venkovní teplota - Z teploměru před vrátnicí
- Minimální teplota -

Obrázek 15 Ukázka grafu typu Line Chart pro zobrazení náhodných dat a referenční hodnoty

### Teploty



Obrázek 16 Ukázka grafu typu Area Chart pro zobrazení náhodných dat a referenční hodnoty

## 8.2.6 Detaily událostí zařízení s vizualizací logovaných parametrů

K logům zařízení se jako uživatelé můžeme dostat pomocí odkazu „Zařízení“ v menu „Sítě“, dále vybrat zobrazení detailů o zařízení a přejít na stránku „Logs from device“. Správce se k logům může dostat přes menu „Administrace / Boiler“, kde vybere zařízení v seznamu zpráv o událostech.

Na stránce je tabulka se seznamem všech doposud zaznamenaných událostí a data, která k nim byly přiložené ve formátu „klíč => hodnota“. Pokud klíč je definován v seznamu parametrů, provede se jeho automatický překlad. U stavových parametrů se převádí i hodnoty na definované překlady stavů.

Pod tabulkou je jednoduchý formulář sloužící pro zobrazení definovaných grafů. Stačí pouze zadat datum, do kterého se data mají vyhledat a délku intervalu, který se má zobrazit. Vyjma denního intervalu probíhá před vykreslením výpočet průměrné hodnoty po časových úsecích. Rozlišení grafu je ve vertikální ose maximálně 600px, pokud by každý pixel představoval jednu hodnotu, mohlo by se zobrazit maximálně 600 hodnot. Existují možnosti použít přiblížení grafu. V případě ročního nebo měsíčního intervalu by se ale přenášelo příliš velké množství dat, mohlo tak by dojít k selhání nebo zpomalení vykreslení grafu. Intervaly jsou celkem čtyři:

- **Denní** – zobrazení všech logů vybraných parametrů za jeden den. Při použití intervalu záznamu dat z kotle každých pět minut, za den bude třeba zobrazit 288 hodnot, není tak nutné provádět průměrování
- **Týdenní** – zobrazení hodnot za posledních sedm dní. Při vzorkování v pětiminutových intervalech by bylo potřeba zobrazit 2 016 hodnot, data se tak průměrují po hodinách, takže výsledných hodnot je jen 168
- **Měsíční** – zobrazení hodnot z intervalu 31 dnů vzorkovaný po tří-hodinových intervalech (pro omezení množství záznamů z 8 928 na 248)
- **Roční** – zobrazení dat z 365 dnů do zadaného data se vzorkovacím intervalem 24 hodin pro snížení počtu záznamů z 105 120 na 365

## Device messages: Zlín Kotel

[Remove all](#)

Celkem položek: 1

Datum	Event name	Serial number	Data	Akce
2014-03-13 06:23:03	Start	1	Kotel 1 => 1, Kotel 2 => 2, event_param_3 => 3, event_param_4 => 4, event_param_5 => 5,	

« předchozí **1** další »

20

Zobrazit Graf

<b>Graph:</b>	Teplota mimo halu <input type="button" value="v"/>
<b>Interval:</b>	Week (sample time: 1 hours) <input type="button" value="v"/>
<b>Do dne:</b>	2014-05-04

Obrázek 17 Výpis událostí ze zařízení a formulář pro zobrazení grafu

## ZÁVĚR

Po zvážení možností uplatnění již navržených monitorovacích systémů s otevřeným zdrojovým kódem byl vybrán systém Freenetis, na kterém byla následně postavena praktická část této práce. Systém Freenetis sice nenabízí natolik propracované nástroje pro sledování síťových zařízení, jako je tomu u některých konkurenčních nástrojů, na druhé straně ale umožňuje evidovat topologii počítačových sítí, poskytuje velmi pokročilou správu přístupových práv uživatelů a díky použité moderní architektuře MVC umožňuje snadnou implementaci nových funkcí. Tyto vlastnosti jsou pro centrální monitoring kotlů, umístěných v sítích zákazníků, velmi důležité. Zákazníci totiž mají vlastní politiku správy sítě, tj. každý zákazník může mít jinak řešen způsob připojení kotlů do své LAN sítě, může mít jiné rozsahy IP adres, pravidla firewallu i poskytovatele připojení k Internetu. Každý zákazník může mít vlastní správce sítě i správce kotlů s různými přístupovými právy.

Verze systému Freenetis, která byla pro tento projekt použita, není v současnou dobu ještě veřejně vydána. Po dohodě s vývojáři byla získána nejnovější verze systému s odlišným nastavením databáze využívající MySQL funkce table partitioning. Tím se má předcházet přetěžováním databáze a serveru v případě vysokého nárůstu záznamů přijatých z jednotlivých kotlů.

Aplikace Freenetis byla rozšířena o funkcionalitu řešící danou problematiku, avšak nemohla být řešena jako přídatný modul, jelikož to návrh aplikace neumožnil. Řešení je navrženo tak, aby zvládlo pokrýt všechny nároky současně realizovaného a testovaného software kotlů navrženého Ing. Romanem Červeňanem. To obnáší pouze logování událostí a maximálně pěti parametrů s událostí spjatých. Samotná realizace webové aplikace je dimenzovaná tak, aby bylo možné rozšiřovat počty událostí a jejich parametrů bez omezení. Je možné i přijímat a zobrazovat parametry ze zařízení, které nejsou spjaté s událostmi v zařízení a slouží pro monitorování otopného systému a jeho prostředí. Těchto parametrů může být neomezené množství. V kapitole **Chyba! Nenalezen zdroj odkazů.** je navržen nový způsob komunikace pomocí HTTP protokolu, který by měl odbourat všechna dosavadní omezení na způsob předávání dat a jejich množství. Přenos dat není šifrován, ale k aktivaci šifrování stačí upravit algoritmus kotle, aby vyvolával zabezpečené požadavky, kterým se aplikace přizpůsobí.

Úspěšně se podařilo splnit všechny body zadání a systém je připraven k testování na konkrétních zařízeních.

## ZÁVĚR V ANGLIČTINĚ

After considering the possibilities of using other monitoring systems with open source code, we have chosen Freenetis as the base for practical part of this work. Although Freenetis does not offer as sophisticated tools for monitoring network devices as it is in the case of competitive tools, it allows to capture various network topologies, provides advanced access rights management and thanks to modern MVC architecture, it is easily extendable with new functionalities.

The version of Freenetis, which was used for this project, is not currently publicly released yet. In an agreement with the developers, a newer version of the system with the MySQL database table partitioning feature was used. Table partitioning prevents overloading of the database and server in case of high volumes of data records received from individual boilers.

Freenetis has been extended with functionality dealing with a specific issues of boiler management.. The solution is designed to cover all the requirements of boiler firmware designed by Roman Červeňan. This includes logging of events and up to five parameters associated with the event. The current implementation of the Web application is designed in a way that it is possible to expand the number of events and their parameters without restrictions. It is also possible to receive and display data from devices that are not connected with the events in the device. This can be used for monitoring the heating system and its environment. There can be unlimited quantities of these parameters. In section 7.2, there is a specification draft of new way to communicate via HTTP protocol, which should eliminate all existing restrictions on data transmission method and quantity. Data transfer is not encrypted, but to activate the encryption, it is sufficient to modify the algorithm of the boiler to induce secure requirements to which the application adapts .

**SEZNAM POUŽITÉ LITERATURY**

- [1] DULÍK, Tomáš, Jan DUBINA, Ondřej FIBICH, Michal KLIMENT a David RAŠKA. *Freenetis: Informační systém pro neziskové sítě* [online]. 2014 [cit. 2014-04-16]. Dostupné z: <http://www.freenetis.org/>
- [2] Instalace ze zdrojových kódů. *Freenetis* [online]. 2014, 2014-01-19 [cit. 2014-05-04]. Dostupné z: [http://wiki.freenetis.org/index.php/Instalace\\_ze\\_zdrojovych\\_kodu](http://wiki.freenetis.org/index.php/Instalace_ze_zdrojovych_kodu)
- [3] Partitioning. *MySQL: The world's most popular open source database* [online]. 1997, 2014 [cit. 2014-05-04]. Dostupné z: <https://dev.mysql.com/doc/refman/5.1/en/partitioning.html>
- [4] SIGMATEK GMBH & CO KG. Sigmatek: Complete automation solutions for machine building [online]. 2014 [cit. 2014-05-05]. Dostupné z: <http://sigmatek-automation.us/>
- [5] SIGMATEK. *LASAL: Operating system* [PDF]. 2004, 983 s., 13.12.2011 [cit. 5.5.2014]. Dostupné z: <ftp://ftp.wexoe.dk/download/Industri/Sigmathek%20tr%E6nings%20manualer/LASAL%20OS.pdf>
- [6] Dohledový systém Zabbix: Představení. *Linuxsoft* [online]. 11.2.2013 [cit. 2014-05-07]. Dostupné z: [http://www.linuxsoft.cz/article.php?id\\_article=1963](http://www.linuxsoft.cz/article.php?id_article=1963)
- [7] ORTEX. *Nagios: Monitorovací a dohledový systém* [online]. 2009, 2014 [cit. 2014-05-07]. Dostupné z: <http://www.nagios.org/>
- [8] ZABBIX SIA. *Zabbix: The Enterprise-class Monitoring Solution for Everyone* [online]. 2001, 2014 [cit. 2014-05-07]. Dostupné z: <http://www.zabbix.com>
- [9] KOHANA TEAM. *Kohana: The swift PHP framework* [online]. 2007, 2014 [cit. 2014-05-08]. Dostupné z: <http://kohanaframework.org/>
- [10] *Apache Server 2: Kompletní příručka administrátora*. Brno: Computer Press, a. s., 2004. ISBN 80-251-0319-6.
- [11] *Mistrovství v PHP 5*. Brno: CP Books, a. s., 2005. ISBN 80-251-0799-X.
- [12] O spalování tuhých paliv v lokálních topeništích: aneb palivo, tvorba znečišťujících látek a spalování jako vztah muže a ženy. TOPINFO S.R.O. *Technické zařízení budov* [online]. 28.5.2012 [cit. 2014-05-10]. Dostupné z: <http://energetika.tzb-info.cz/8644-o-spalovani-tuhych-paliv-v-lokalnich-topenistich-2>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

Apache	Virtuální webový server umožňující správu domén a jejich dat.
fastCGI	Protokol umožňující rychlejší komunikace mezi serverem a klientem.
GitHub	Hostingový projekt pro správu a vývoj softwarových produktů.
GNU	Svobodný software inspirovaný operačními systémy typu unix.
GPL	(General Public Licence) Licence pro svobodný software.
HTTP	(HyperText Transfer Protocol) Webový protokol pro přenos dat při klient-server komunikaci založený na principu požadavku a odpovědi.
HTTPS	Jedná se o HTTP protokol zabezpečený pomocí vrstvy SSL.
InnoDB	Typ databázového uložení umožňující vytvářet relace mezi tabulkami.
Kohana	PHP framework s otevřeným zdrojovým kódem pro vývoj webových projektů, na kterém je postaven i systém Freenetis
MAC	(Media Access Control) Unikátní identifikátor síťového zařízení
MVC	(Model View Control) Softwarová architektura oddělující logiku správy dat(Model) a vykreslení (View) pomocí řídicí logiky(Control)
MySQL	Multi-platformní databázový systém
PHP	Skriptovací programovací jazyk určený zejména pro webové aplikace
ORM	(Object Relation Mapping) Nástroj převádějící databázové data na objektové struktury umožňující snadnější vývoj a práci s databází.
SourceForge	Hostingový projekt pro komunitní vývoj softwarových produktů.
SSH	(Secured SHell) Služba vytvářející tunelové šifrované spojení mezi klienty.
SSL	(Secured Socket Layer) Vrstva pro zabezpečení komunikace mezi klienty navržená pro protokoly HTTP, POP3, IMAP4, XMPP a FTP.
VoIP	(Voice over Internet Protocol) Technologie umožňující přenos zvuku pomocí Internetových protokolů UDP/TCP/IP.

**SEZNAM OBRÁZKŮ**

Obrázek 1 Graf závislosti koncentrace složek ve spalinách a účinnosti na množství vzduchu při spalování pevných paliv [12].....	12
Obrázek 2 Webový instalační formulář aplikace Freenetis.....	28
Obrázek 3 Vybrané tabulky a jejich relace na databázové vrstvě.....	35
Obrázek 4 Závislosti tabulek pro uchování logů parametrů a událostí.....	37
Obrázek 5 Ukázka tabulky s fixními parametry .....	38
Obrázek 6 Tabulky parametrů, grafů a jejich závislostí.....	40
Obrázek 7 Náhled na detaily o zařízení typu kotel .....	42
Obrázek 8 Formulář pro vložení nové šablony zařízení.....	43
Obrázek 9 Formulář pro přidání nového zařízení (krok 1).....	44
Obrázek 10 Formulář pro přidání nového zařízení (krok 2).....	45
Obrázek 11 Definice nového stavového parametru .....	46
Obrázek 12 Výstřižek stránky s výpisem parametrů.....	47
Obrázek 13 Nastavení zobrazení grafů pro stavové parametry .....	48
Obrázek 14 Ukázka koláčového grafu stavu zařízení pro jeden stavový parametr .....	48
Obrázek 15 Ukázka grafu typu Line Chart pro zobrazení náhodných dat a referenční hodnoty .....	49
Obrázek 16 Ukázka grafu typu Area Chart pro zobrazení náhodných dat a referenční hodnoty .....	49
Obrázek 17 Výpis událostí ze zařízení a formulář pro zobrazení grafu .....	51

**SEZNAM TABULEK**

Tabulka 1 Rozšiřující záznamy databázové tabulky enum_types .....	41
Tabulka 2 Rozšíření databázové tabulky axo o nové záznamy .....	41

## SEZNAM PŘÍLOH

Přílohy jsou uloženy v elektronické podobě na CD. Součástí jsou přiloženy zdrojové kódy aplikace včetně návrhu databáze a elektronické formy bakalářské práce ve formátech doc a pdf.