Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Systém pro administraci IoT Wi-Fi sítě

Plzeň 2023 Matěj Rampula

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Matěj RAMPULA**
Osobní číslo:	**A19B0178P**
Studijní program:	**B0613A140015 Informatika a výpočetní technika**
Specializace:	**Informatika**
Téma práce:	**Systém pro administraci IoT Wi-Fi sítě**
Zadávající katedra:	**Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s prostředím Wi-Fi sítí na ZČU a použitým ověřovacím systémem.
2. Navrhněte systém pro správu registrací a administraci připojování IoT zařízení do Wi-Fi sítě s využitím technologie IPSK (Identity PSK).
3. Zohledněte potřebu vytváření skupin zařízení a ovládání jejich komunikace.
4. Navržený systém realizujte jako modulární virtuální stroj.
5. Řešení ověřte sadou netriviálních testů.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce

Vedoucí bakalářské práce: **Ing. Martin Šimek, Ph.D.**
Centrum informatizace a výpočetní techniky

Datum zadání bakalářské práce: **3. října 2022**
Termín odevzdání bakalářské práce: **4. května 2023**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 25. října 2022

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 22. června 2023

Matěj Rampula

Abstract

The goal of this thesis if to design and implement a Proof of Concept system, that will allows a user-friendly connecting and disconnecting of differenct devices to the WiFi network of the University of West Bohemia. The system will be designed to let the user aggregate the devices into groups; each group will have a iPSK key that will be used to access the WiFi network. This thesis aims to elevate quality of life for teachers and to facilitate their work.

Abstrakt

Obsahem práce je návrh a implementace formou Proof of Concept systému, který umožní uživatelsky snadné připojování a odpojování zařízení do WiFi sítě Západočeské univerzity. Systém umožní sdružování koncových zařízení do skupin, které si definuje uživatel. Bude použita technologie Cisco iPSK, kdy každá skupina bude mít přidělený vlastní klíč pro přístup do sítě. Cílem je usnadnění práce (nejen) vyučujících a zlepšení kvality výuky.

Obsah

1 Úvod

Pokud chce dnes vyučující používat ve svém předmětu zařízení s přístupem do WiFi sítě ZČU, musí si vymyslet vlastní řešení jejich připojení, sám si ho vytvořit a nakonfigurovat. To znesnadňuje práci zejména méně technologicky zdatným, ale i těm zkušeným, kteří by svůj čas mohli naplnit lépe.

Cílem této práce je navrhnout a vytvořit jednotný systém, který bude umožňovat jednoduché připojování a odpojování zařízení do sítě. Bude dovolovat připojení do veřejného internetu (například pro potřeby výuky o počítačových sítích, nebo aby mohli studenti na projektu pracovat i z domova), snadnou správu zařízení podle uživatelem definovaných skupin, přidělování hostname a statických IP adres. Věřím, že všechny tyto funkcionality jsou žádané a budou užívané.

Práce je rozdělena na část návrhu systému, ve které se pozornost věnuje tomu, jak by systém měl na infrastruktuře školy fungovat, a na část programátorské dokumentace, kde je popsána implementace systému jako Proof of Concept.

Čtenář by měl být seznámen se základní terminologií vývoje ve webovém prostředí, se strukturou počítačových sítí a ideálně i s technologií Cisco, aby si odnesl co nejvíce.

2 Návrh systému

Stavba portálu je pojata jako, pro webové aplikace klasický, vzor architektury software Model-View-Controller (MVC). Zvlášť budou tedy definovány třídy pro logické řízení, správu dat a uživatelské rozhraní. Jelikož je aplikace navrhována s myšlenkou co nejvíce zjednodušit práci uživatele (a odstínit jej tak od složitostí vnitřního fungování serveru ISE; počítá se s používáním takzvanými „OMZU“, tj. obyčejnými muži z ulice), bude způsob zpracování logiky a propojení datového modelu lokálně skladovaných dat a těch získaných z ISE pravděpodobně místy trochu složitější, ale dostatečně kompromisní.

Kromě MVC portálu bude součástí aplikace fronta zpráv a workeři, kteří je budou zpracovávat. Bude se tedy jednat o mikroservisní architekturu. Pomocí portálu bude probíhat interakce s uživatelem a mikroservisy budou provádět bezpečnostně kritické operace, například manipulaci s firewallem. Kromě většího zabezpečení nabízí toto řešení i větší spolehlivost; ukládání zpráv umožní obnovení např. při update systému nebo v případě havárie.

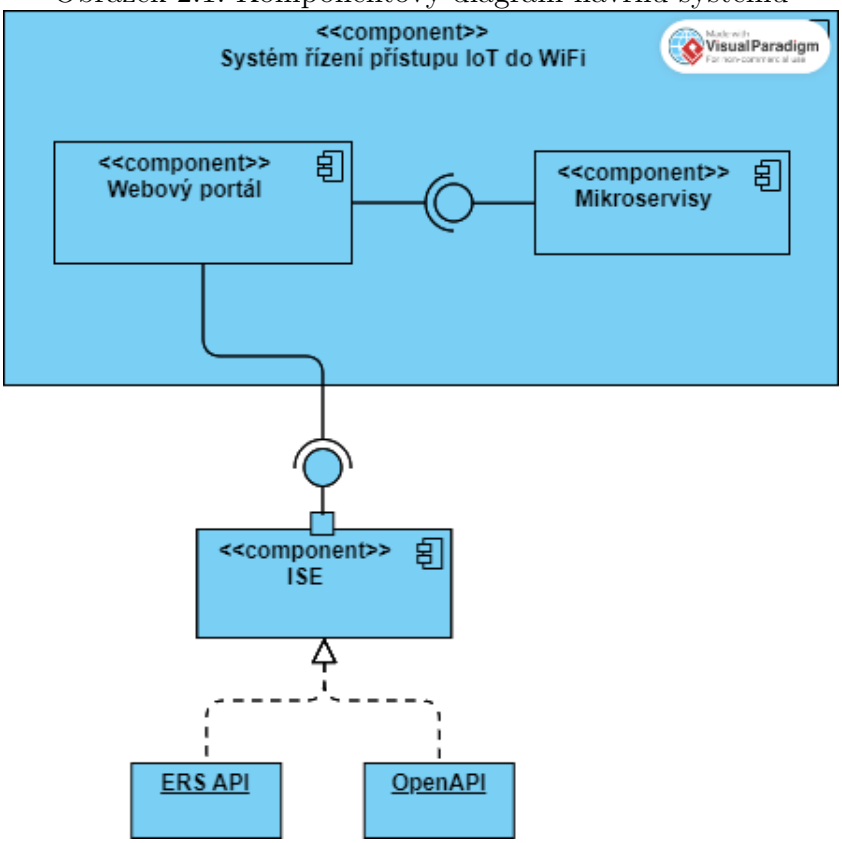
Další komponentou systému bude DHCP server. Ten bude umožňovat statické přidělování IP adres dle přání uživatele – např. bude možné mít pro dvacet zařízení používaných při výuce současně blok dvaceti po sobě jdoucích IP adres. Při realizaci bude použito již hotové řešení, jelikož tvorba vlastního DHCP serveru by byla náročná a zbytečná. Čtení stavu bude probíhat přes databázi a změna parametrů serveru se bude odehrávat pomocí mikroslužeb, pro již výš zmíněné výhody.

Poslední součástí systému bude DNS server. Důvodem k jeho užití je možnost přidělení člověkem čitelných a zapamatovatelných řetězců jednotlivým zařízením, opět dle vůle uživatele. Při pokračování příkladu s DHCP: pro dvacet současně používaných zařízení bude možné přidělit jména typu „arduino-01“ až „arduino-20“. Toto usnadní práci i koncovým uživatelům, například pokud budou chtít, aby mohlo jejich zařízení komunikovat se zařízením spolužáka. Čtení bude opět uskutečněno pomocí přímého přístupu do databáze a konfigurace DNS skrze mikroslužby.

2.1 Propojení s ISE

Jak bylo naznačeno výše, data budou skladována lokálně v databázi, aby bylo dosaženo co nejlepší latence při přistupování k nim a při tvorbě odpovědi pro uživatele. Databáze bude plněna buď přímo z aplikace portálu (např.

Obrázek 2.1: Komponentový diagram návrhu systému



při vytváření nových entit, kdy jsou známy všechny potřebné položky), nebo daty získanými z API Identity Services Engine (ISE) serveru. Místní data budou ukládána v poměru 1:1, tj. jeden logický objekt v ISEodpovídá jednomu řádku v jedné tabulce v databázi. Toto povede ke snazší komunikaci s ISE, ale zvýší to nároky na logickou stránku aplikace portálu kvůli zjednodušení pro uživatele – uživatel například nepotřebuje vědět, že při přidělení nebo odebrání přístupu do internetu dochází ke změně skupiny koncových zařízení; stačí mu zobrazit checkbox.

API a komunikace s ISE bude dále popsáno v sekci 3.2.5.

2.2 Základní prvky

Dle zadání byl při navrhování kladen důraz na dvě věci: potřebu vytváření skupin zařízení a ovládání jejich přístupu do internetu. Skupiny zařízení jsou nezbytné pro přehlednost a logické rozdělení: uživatel chce mít při výuce jednoho předmětu jednu sadu zařízení, která ho nezajímají při výuce předmětu jiného. Ovládání přístupu do internetu zaručí vyšší bezpečnost pro zařízení

i obecně pro síť školy. Vzhledem k těmto parametrům jsou brány jako základní prvky koncové zařízení (dále také Endpoint) a skupiny koncových zařízení (dále také EndpointGroup). Dalšími prvky jsou autorizační profily (AuthorizationProfile), autorizační pravidla (AuthorizationRule) a sada politik (PolicySet). Anglické ekvivalenty odpovídají terminologii ISE.

2.2.1 Koncové zařízení

Každý Endpoint reprezentuje jedno koncové zařízení, což je pro naše účely cokoli, co má MAC adresu, protože ta je brána jako unikátní identifikátor. Dalšími vlastnostmi koncového zařízení jsou jméno, IP adresa, příslušnost ke skupině a přístup do internetu. Jméno bude využíváno pro usnadnění orientace v portálu a jako hostname přidělený přes DNS server. IP adresa bude moci být dynamická i statická (ovládané přes DHCP server). Jedno koncové zařízení může být kvůli omezením ISE v právě jedné skupině (tedy 1:1)

2.2.2 Skupina koncových zařízení

Jedna skupina obsahuje N koncových zařízení. Vlastnostmi skupiny jsou tedy seznam zařízení, potom jméno a popis. Popis i jméno jsou čistě pro uživatele, jméno ovšem může (opět kvůli omezením ISE) obsahovat pouze písmena, číslice, podtržítka a pomlčky, což bude muset být zohledněno v implementaci. Bude určena defaultní skupina „Unknown“, do které spadnou zařízení, která nebudou náležet do žádné skupiny. Zařízení v defaultní skupině nebudou moci přistupovat do veřejného internetu.

Ve vnitřním mechanismu budou za každou skupinu, kterou vidí uživatel, dvě. Před jméno zvolené uživatelem bude jako prefix přidán název SSID a za jméno suffix vypovídající o přístupu do sítě – buď „-internet“, pokud budou zařízení moci do veřejného internetu, nebo „-lan“, pokud budou moci jen do lokální sítě. Tento systém je zvolený kvůli podmínkám autorizačních pravidel, což bude rozvedeno dále.

2.2.3 Autorizační profil

Pro každou vnitřní skupinu je vytvořen autorizační profil se stejným jménem jako má skupina. K profilu je přiřazeno iPSK, tedy klíč užitý při připojování zařízení do sítě, Downloadable Access Control List (DACL), Airespace ACL a role na kontroleru. DACL bude v ISE vytvořený a aplikace s ním nebude nijak manipulovat, stejně jako s konfigurací kontrolerů. Bude existovat výchozí autorizační profil, který umožní přístup pouze do lokální sítě.

2.2.4 Autorizační pravidlo

Pro každou vnitřní skupinu bude vytvořeno také autorizační pravidlo, které při splnění přidělí příslušný autorizační profil. Pravidlo bude testovat příslušnost zařízení ke skupině koncových zařízení. Kontrolovaná podmínka bude atribut „Name“ ze slovníku „IdentityGroup“. Bude existovat pravidlo, které zařízení přidělí výchozí autorizační profil.

2.2.5 Sada politik

Sada politik sdružuje autorizační pravidla. Pro každé SSID je v ISE vytvořena právě jedna sada. K jejímu vytvoření dojde při instalaci portálu. Pro vybrání sady politik ISE bude testovat položku „Radius · Called-Station-ID“, tedy z jakého SSID koncové zařízení přistupuje.

Napevno nastavené bude autentizační pravidlo, které pustí zařízení k dalšímu zpracování, pokud nebude mít přiděleno nějakého registrovaného uživatele.

2.3 Návrh uživatelského rozhraní

Uživatelské rozhraní, tj. webová stránka, se bude skládat z domovské obrazovky, správy koncových zařízení, správy skupin a nastavení. Na domovské obrazovce budou zobrazena právě připojená zařízení pro rychlý přístup (například aby uživatel nemusel dlouho hledat, pokud se student zeptá, jakou IP adresu má jeho zařízení). Také na ní bude přehled skupin s počtem aktivních zařízení z celkového počtu ve skupině.

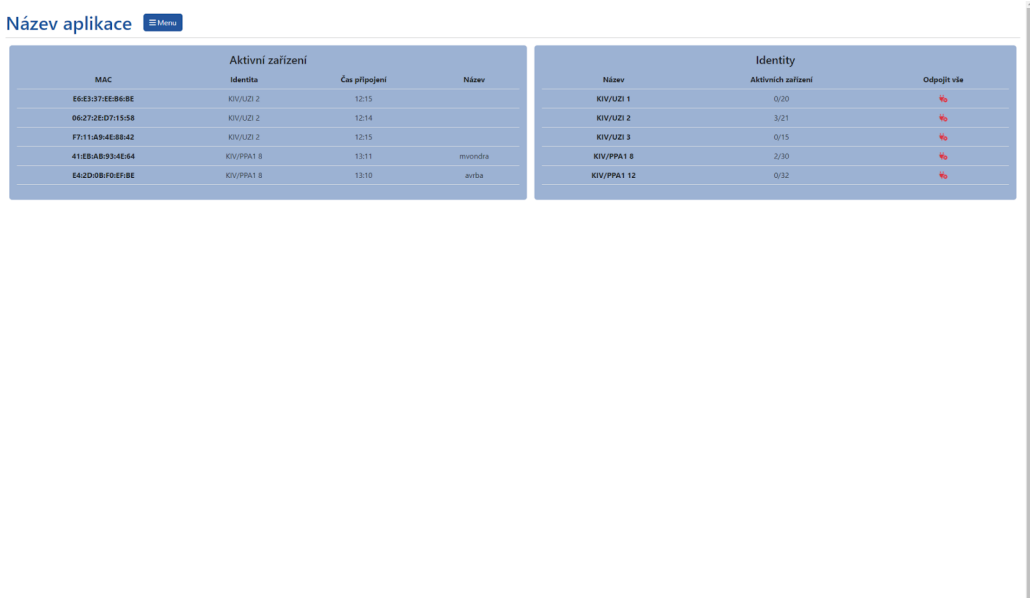
Na stránce správy koncových zařízení bude přehled všech registrovaných (i neregistrovaných, pokud o nich bude vědět ISE) zařízení. V přehledu budou vypsány všechny jejich vlastnosti. Také bude možnost přidat nové zařízení a upravit nebo smazat stávající. Přidávání a upravování bude probíhat asynchronně a formulář bude zobrazován v modální stránce, protože mi přijde, že takový přístup méně narušuje tok myšlenek – pokud jsem na stránce, kde chci spravovat zařízení, dává smysl, že z ní kvůli tomu nemusím odejít na stránku s formulářem. Posledně bude existovat možnost obnovit seznam, kdyby například došlo k připojení nového zařízení.

Koncová zařízení budou seskupena podle skupin, do nichž patří. Ačkoli vnitřně existují skupiny pro přístup do internetu a pro přístup do lokální sítě, uživateli se bude zobrazovat pouze jedna skupina s agregovanými zařízeními obou vnitřních. Bude možné rozbalovat a sbalovat skupiny.

Na stránce správy skupin koncových zařízení bude přehled všech skupin a jejich vlastností. Bude existovat možnost přidat novou skupinu a upravovat a mazat stávající. Úpravy a přidávání budou opět probíhat v modální stránce s formulářem.

V nastavení bude možnost měnit základní parametry aplikace, například přístup k ISE, k databázi nebo SSID. Neočekává se změna těchto parametrů, ale je dobré být připraven.

Obrázek 2.2: Počáteční návrh domovské obrazovky



2.4 Flow akcí

Při navrhování bylo uvažováno zejména několik základních akcí: vytváření skupiny, vytváření koncového zařízení, změna přístupu do internetu. Kromě těchto akcí se samozřejmě počítá s dodatečnými změnami a mazáním koncových zařízení i skupin.

2.4.1 Vytváření skupiny zařízení

Pro uživatele je vytvoření skupiny otázkou vyplnění jména a iPSK. Na pozadí dojde k vytvoření (a uložení do databáze) dvou skupin koncových zařízení (jedna s přístupem k internetu a jedna s přístupem do lokální sítě). Messengeru bude předána žádost o vytvoření skupin v ISE. Messenger žádost uloží a předá správnému ISEHandleru, který se postará o odeslání POST požadavku na API ISE serveru. Dále budou vytvořeny autorizační profily s

uživatelem určeným iPSK a výchozími hodnotami DACL, airespace ACL a tak podobně. Požadavky na jejich vytvoření budou také přes Messenger a ISEHandler odeslány ISE serveru. Posledně dojde k vytvoření autorizačních pravidel – opět jedno pro přístup k internetu a jedno pro přístup k lokální síti – a jejich tvorbě v ISE (počítá se s tím, že sada politik je již vytvořená). Tím bude vytvořeno vše pro plánovanou funkci systému a skupiny se budou moci plnit zařízeními.

Počítá se také s hromadným tvořením skupin; uživatel nahraje CSV soubor, kde jeden řádek bude odpovídat jedné skupině. Server soubor naparsuje a vytvoří požadované skupiny.

2.4.2 Vytváření koncového zařízení

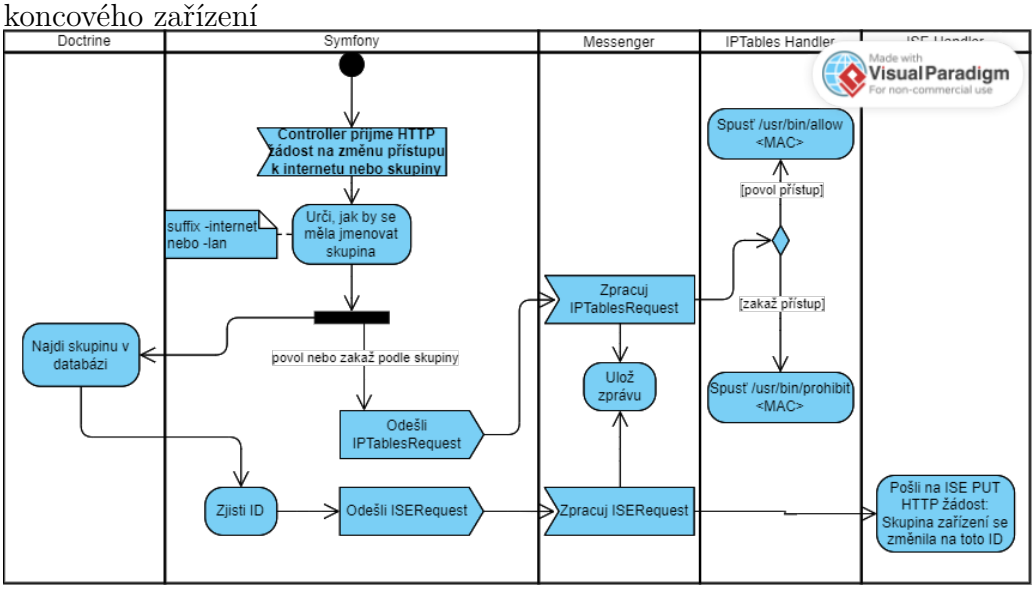
Uživatel odešle formulář se jménem, MAC adresou, IP adresou, příslušností ke skupině a logickou hodnotou přístup/nepřístup k internetu. V backendu bude podle přístupu k internetu a kořene jména skupiny (tj. bez prefixu SSID a sufixu) nalezena v databázi správná skupina, vytvořena instance koncového zařízení a uložena do databáze. Přes Messenger bude předána žádost o vytvoření koncového zařízení s příslušnou skupinou na ISE server. Dále bude přes Messenger předána žádost o změnu konfigurace DHCP a DNS serveru a příslušné handlery změny provedou. Kvůli případným kolizím při přidělování hostname bude nutné mít unikátní název zařízení nebo k neunikátnímu přidat celé číslo tak, aby se unikátním stal. Nakonec se přes Messenger předá žádost IPTablesHandleru, který provede změny ve firewallu a zakáže/povolí připojení do internetu.

Opět se počítá s hromadným tvořením koncových zařízení pomocí CSV souboru.

2.4.3 Změna přístupu do internetu

Pokud bude chtít uživatel zakázat nebo povolit přístup do internetu nějakého koncového zařízení, změní při upravování zařízení zaškrtávací políčko a odešle na server. Ten podle kořene jména skupiny a podle nové hodnoty přístupu k internetu nalezne správnou skupinu a přes Messenger odešle žádosti IPTablesHandleru a ISEHandleru. IPTablesHandler změní firewall – povolí nebo zakáže přístup k internetu. ISEHandler metodou PUT odešle na ISE server žádost o změně skupiny zařízení. Tato akce je zobrazena v obrázku 2.3.

Obrázek 2.3: Diagram aktivit při změně skupiny nebo přístupu do internetu koncového zařízení



2.5 Kontrola přístupu

Uživatelé se budou do portálu přihlašovat skrze jednotné přihlašování ZČU, tedy pomocí Orion účtu. Pro zajištění pouze autorizovaného přístupu bude využita komponenta Symfony Firewall. Není žádoucí, aby si studenti (resp. jiní vlastníci koncových zařízení) sami měnili konfiguraci přístupu do sítě. Přihlášení do portálu bude tedy umožněno pouze učitelům nebo osobám, které o to požádají CIV a jejichž odůvodnění bude shledáno vhodným. Ověřování role bude probíhat pomocí příslušnosti do skupin v LDAP; k získávání informací z LDAP nabízí Symfony službu LDAP User Provider. Uživatel bude moci měnit pouze záznamy, které sám vytvořil.

2.6 Síť

Server, na němž běží portál, bude mít dvě síťová rozhraní. Jedno s IP adresou z privátního adresního rozsahu 10.0.0.0/8 a druhé s IP adresou s přístupem do sítě školy. Na privátním rozhraní bude server mít roli výchozí brány pro zařízení ve WiFi síti s jedním SSID. Koncová zařízení, která mají mít přístup do veřejného internetu, budou forwardována na druhé rozhraní, proběhne překlad zdrojové adresy na veřejnou (SNAT) a dále budou moci přistupovat do internetu.

2.7 Firewall

Řízení přístupu koncových zařízení do veřejného internetu bude ovládáno pomocí firewall serveru, na němž běží portál. Pokud budou mít přístup zakázaný, spadnou do defaultní policy REJECT v řetězci FORWARD. Pokud budou mít přístup povolený, bude v řetězci HUB existovat pravidlo, které jim přístup dále umožní. Do řetězce HUB se bude skákat z řetězce FORWARD.

2.8 Zvolené technologie

Pro realizaci návrhu byl vybrán MVC framework Symfony, který je napsaný v programovacím jazyce PHP a obsahuje množství hotových komponent. Konkrétně Symfony bylo zvoleno proto, že jsem s ním již měl zkušenosti z osobních projektů. Kromě backendu v PHP je Symfony připraveno i na budoucí použití JavaScriptu pro vylepšení a rozšíření uživatelského frontendu. Symfony bude použito pro tvorbu hlavního backendu i mikroslužeb, jelikož nabízí komponentu Messenger, která je právě na mikroservisní architekturu stvořená.

V Symfony je také zabudovaná komponenta Doctrine, což je object-relational mapper (ORM). Nabízí jednoduché převádění objektů objektově orientovaného programování do relací používaných v databázích. [1]

Jako systém řízení báze dat bylo zvoleno open-source PostgreSQL. Nabízí všechny požadované funkce, je mi známé a, ačkoli je méně populární než MySQL a nepatrně náročnější na obsluhu, bylo vyhodnoceno jako lepší varianta. [4] Podporuje množství datových typů (mimo jiné i JSON) a je připravené na náročnější dotazy, kdyby v budoucnu vyvstala potřeba složitějších analýz.

Pro funkci DHCP serveru byla vybrána Kea DHCP. Její hlavní výhody jsou, že se jedná o open-source implementaci DHCP serveru, tedy je zdarma, že umožňuje čtení svého stavu (například zapůjčených IP adres) přímo z databáze, a že po změně konfigurace nevyžaduje restart. Změna nastavení je také možná přes REST API, což více umožňuje využívat strukturu systému a provádět konfiguraci pomocí mikroslužeb. [2]

Do role DNS serveru byl určen PowerDNS. Stejně jako Kea splňuje požadované funkce, tedy čtení stavu z databáze a využití mikroslužeb. Je open-source a pro rozsah projektu více než dostačující. [3]

2.9 Nasazení

Systém bude vytvořený jako Docker kontejner, aby bylo umožněno co nejsnazší nasazení. Při prvním spuštění proběhne instalační skript, který uživatele provede nastavením hlavních parametrů, jako jsou SSID sítě, kterou má systém ovládat, připojení do PostgreSQL a připojení na ISE server. Po nastavení skript provede prvotní úkony typu vytvoření sady politik pro dané SSID a získání dat z ISE a jejich uložení do databáze.

Asi by bylo možné (a žádoucí) zahrnout všechny používané technologie do jednoho kontejneru, avšak to je prostor pro budoucí vylepšování. Prozatím se počítá s tím, že portál, PowerDNS, Kea DHCP i PostgreSQL budou mít každý svůj kontejner a budou spolu komunikovat pomocí vnitřních sítí Dockeru. PowerDNS a Kea DHCP (a samozřejmě portál) budou mít přístup na soukromé rozhraní serveru (s rozsahem adres 10.0.0.0/8).

Systém bude nasazován na virtuální server ve službě OpenNebula, která je v prostředí ZČU dobře zavedená.

3 Programátorská dokumentace

Pro účely Proof of Concept je prvotní implementace provedena jednodušším způsobem. Mikroservisní architektura je zatím nahrazena monolitickou stavbou, budoucí změna zpět k plánované struktuře však bude jen záležitostí vytvoření vrstvy navíc.

3.1 Adresářová struktura

V kořenovém adresáři projektu jsou tyto složky:

- assets – soubory týkající se frontendu,
- bin – spustitelné soubory Symfony,
- config – všechny konfigurační soubory,
- migrations – migrace Doctrine,
- node_modules – stažené soubory Node.js,
- public – veřejné soubory vystavené na internet; vstupní bod portálu,
- src – veškerý kód PHP,
- templates – šablony Twig,
- tests – testy,
- var – dočasné soubory, cache, logy,
- vendor – stažené soubory Composeru.

3.2 Symfony

Jak již bylo uvedeno výše, Symfony je Model-View-Controller (MVC) framework s bohatou zásobou komponent. Pro správu těchto komponent využívá Composer, což je manažer knihoven a jejich závislostí pro jazyk PHP. Nejdůležitějšími komponentami pro tento projekt jsou: Doctrine, Form, Messenger, Security, Twig a Webpack Encore.

Jednou z dobrých schopností Symfony je Autowiring. Pokud je nějaká třída definovaná jako služba (Service; podle výchozí konfigurace jsou jako služby definované všechny třídy ve složce `src`), je možné použít v jejich konstruktoru a metodách Dependency Injection. Pokud například bude signatura metody `public function index(EndpointRepository $endpoint_repository): Response`, znamená to, že Symfony při zavolání automaticky vloží požadované parametry, v tomto případě repozitář koncových zařízení. Při autowiringu se argumenty vkládají podle napovídání typu (type-hinting) i podle názvu argumentu. Všechny možnosti vypíše příkaz `php bin/console debug:autowiring`.

Další šikovnou schopností je využívání anotací. Například cesta k zobrazení koncových zařízení (níže v tabulce `/endpoints`) se definuje takto `#[Route('/endpoints', name: 'app_endpoints')]`. Anotace usnadňují čitelnost kódu, poněvadž umožňují zahrnutí konfigurací do zdrojového kódu místo oddělených souborů.

3.2.1 Hlavní framework

Při příchodu požadavku na server nejprve komponenta Routing naparsuje jeho URL a podle ní určí Controller a metodu, kterou se má požadavek obsloužit. Příslušná metoda provede, co je po ní žádáno (např. najde něco v databázi) a odešle uživateli zpět odpověď (např. tabulku všech koncových zařízení).

Přehled cest

V aplikace jsou definovány tyto cesty (všechny názvy mají ještě prefix „app_“, ale v tabulce byl vynechán kvůli úspoře místa):

Název	Cesta	Význam
configuration	/configuration	Konfigurace serveru
dhcp	/dhcp	Záznamy o zápůjčkách IP adres
endpoint_groups	/groups	Seznam skupin koncových zařízení
endpoint_groups_add	/groups/add	Formulář na přidání skupiny a zároveň jeho příjem
endpoint_groups_edit	/groups/edit/{id}	Formulář na upravení skupiny s ID
endpoint_groups_remove	/groups/remove/{id}	Smazání skupiny s ID
endpoints	/endpoints	Seznam koncových zařízení
endpoint_add	/endpoints/add	Formulář na přidání zařízení a zároveň jeho příjem
endpoint_edit	/endpoints/edit/{id}	Formulář na upravení zařízení s ID
endpoint_remove	/endpoints/remove/{id}	Smazání zařízení s ID
home	/	Domovská obrazovka
ise_refresh	/ise/refresh/{what}	Zavolá ISE API a obnoví záznamy v databázi

Tabulka 3.1: Tabulka s přehledem cest

{what} v cestě `app_ise_refresh` může nabývat hodnot policies, groups, profiles a endpoints.

Doctrine

Doctrine je komponenta, která mapuje objekty do relační databáze. Třídy, které spravuje, se nacházejí ve složce `src/Entity`. Tento adresář je dále rozdělený na `Kea` a `Main`, protože se v projektu pracuje se dvěma připojeními do databáze, jedno pro Kea DHCP a druhé pro vlastní potřeby aplikace. Sloupce v tabulce databáze se definují pomocí anotace `#[ORM\Column]`; primární klíč tabulky určuje anotace `#[ORM\Id]`.

V průběhu vývoje dochází k aktualizaci stavu databáze pomocí výše naznačených migrací (nacházejí se ve složce `migrations`). V produkčním prostředí se se změnami databáze již nepočítá a databáze bude zcela vytvořena při instalaci.

Form

3.2.2 Entity

Pro jednu entitu je v databázi vytvořena jedna tabulka (v případě M:N relace je vytvořena tabulka i pro ni). Změny v kódu se do databáze propíšou příkazy `php bin/console make:migration` a `php bin/console doctrine:migrations:migrate`.

Každá entita má vytvořený repozitář, který spravuje přístup do databáze (volají se přes něj dotazy). Každý repozitář má základní metody typu „najdi vše“, „najdi podle“, „ulož“ nebo „smaž“. Repozitáře entit v namespace \Main implementují rozhraní `IFindyOneById`, které zaručuje možnost najít instanci podle ID, jelikož všechny entity mají ID. Toto rozhraní je použito při volání API. Repozitáře jsou převážně stejné, proto budou komentovány jen v případě speciálních funkcí.

Main\Endpoint

Tato entita reprezentuje koncové zařízení. Obsahuje vlastnosti `$id`, `$name`, `$mac`, `$group` a `$internetAccess`. Věřím, že jejich názvy jsou dostatečně významové. Atribut `$group` je instance třídy `EndpointGroup`, do které zařízení patří. Konstruktor používá parametr id skupiny a hledá si správnou skupinu sám, což odporuje návrhovému vzoru Dependency Injection a bude to v produkčním řešení opraveno.

Main\EndpointGroup

EndpointGroup reprezentuje skupinu koncových zařízení. Má atributy `$id`, `$name`, `$description` a `$endpoints`. Poslední jmenovaný je instancí rozhraní Collection balíčku Doctrine. V konstruktoru se inicializuje na prázdnou kolekci, kterou Doctrine líně naplní, pokud je přistupováno k jejím prvkům.

Protože je uživateli zobrazována jedna skupina zařízení, ačkoliv vnitřně se pracuje se dvěma, je repozitář EndpointGroupRepository složitější. Metoda `findOpposite` nalezne v předaném poli skupinu, která má opačný sufix od té předané, tj. dostane-li skupinu se sufixem „-internet“, pokusí se v poli nalézt skupinu se sufixem „-lan“ a obráceně. Metoda `getJoined` z databáze získá všechny skupiny a postupně je prochází a spojuje – skupinu „Unknown“ nechá beze změny (protože nemá protějšek; obsahuje neznámá zařízení bez přístupu do sítě), pro ostatní skupiny zavolá metodu `findOpposite` a spojí jejich kolekce endpointů do jedné. Při spojování je nutné kolekce inicializovat (tedy naplnit z databáze), což se nejjednodušeji provede převodem na pole a vytvořením nové kolekce. Skupina, která byla jako „opak“ je smazána z pole, nová kolekce je nastavena původní skupině a po projití celého pole skupin je toto pole vráceno.

Další speciální metodou repozitáře skupin je `findOneByRootAndSuffix`. Jejím úkolem je najít skupinu podle kořene (tj. jméno zadané uživatelem, bez prefixu SSID a sufixu s přístupem k internetu) a daného sufixu. Používá SQL výraz `LIKE`.

Čtvrtá specifická metoda je `findAllStripped`. Dělá víceméně to stejné, jako `getJoined`, ale nezaobírá se kolekcemi koncových zařízení, pouze vrátí pole skupin, které mají unikátní kořen.

Main\AuthorizationProfile

Tato entita představuje autorizační profil, tedy výsledek autorizačního procesu v ISE. Místně je ukládáno pouze `$id`, `$name` a `$psk`. Má ještě atribut `$internetAccess`, který se neukládá do databáze a používá se jen při volání API ISE.

Main\AuthorizationRule

AuthorizationRule reprezentuje autorizační pravidlo v sadě politik. Vlastně je to podmínka, která se musí splnit pro přidělení autorizačního profilu. Má atributy `$id`, `$name` a `$profile` a `$groupHierarchy`. Poslední je hierarchie skupiny, které je testované zařízení členem; jelikož jsou námi vytvářené skupiny podskupinami `HubDevices`, jedná se o řetězec formátu ’`Endpoint`

`Identity Groups:HubDevices:<název skupiny>`'.

Main\PolicySet

Poslední entitou vlastní části aplikace je reprezentant sady politik. Má atributy `$id`, `$name` a `$description`. Jeho hlavním účelem je možnost získat z databáze jeho ID pro vytváření nových autorizačních pravidel a úpravu stávajících. Jelikož nynější návrh počítá s jednou instancí portálu na jedno SSID, bylo by možné uložit si ID sady politik do konfigurace, ale dynamické získání z ISE mi kvůli možným změnám přijde vhodnější.

Kromě základních funkcí nabízí repozitář PolicySetRepository specifickou metodu `findPolicyOfThisStation`. Ta najde PolicySet, který odpovídá SSID aktuálního portálu. SSID je uložené jako parametr a proto je v konstruktoru repozitář injektován `ContainerBagInterface`, který obsahuje parametry kontejneru, ve kterém běží Symfony.

Kea\Lease4

Jediným zástupcem z Kea DHCP části aplikace je Lease4. Představuje zápůjčku IPv4 adresy zařízení. Demonstruje možnost připojení se do databáze Kea a budoucí rozšíření aplikace o informace DHCP serveru. Hlavními atributy jsou `$address`, tj. IP adresa uložená v číselné podobě (jako long) a `$hwaddr`, MAC adresa uložená v binární podobě jako proud. Tyto speciálnější způsoby uložení vyžadovaly vytvoření Twig filtrů pro člověkem čitelné zobrazení.

Při práci s entitou je třeba dávat pozor na jiného persistentManagera. Protože je Lease4 načítán z jiné databáze, má logicky i vlastní připojení a Doctrine tedy používá i jiné instance správců uchování.

3.2.3 Controllery

Controllery jsou vstupním bodem kódu, který není součástí frameworku. Nachází se ve složce `src/Controller`. Následuje popis hlavních controllerů a jejich funkcí.

ConfigurationController

Bude sloužit k obsluze nastavování portálu. Zatím není nic implementováno a zobrazuje prázdnou stránku.

DHCPController

Umožňuje zobrazení zápůjček IPv4 adres Kea DHCP serveru. V produkční verzi nebude přítomen, neboť manipulace s DHCP bude probíhat vnitřně a uživatel bude moci IP adresy upravovat přímo u koncových zařízení.

EndpointGroupsController

Tento controller má implementovánu metodu `index` (cesta `/groups`), která zobrazí seznam všech skupin koncových zařízení, a má připravené pahýly pro úpravu a mazání skupin; nebyly ještě implementovány, pro účely Proof of Concept bylo řešení postačující.

Důležitou metodou je `add` (cesta `/groups/add`. Řeší vytváření skupiny a celou kaskádu na to navázaných akcí. Pokud je volána s prázdnou žádostí (autowiring `Request`), pouze zobrazí formulář pro vytvořené nové skupiny. Pokud je volána s vyplněnou žádostí, tedy formulář obsahuje data, dojde k vytváření skupiny:

- nejprve proběhne základní ošetření vstupu uživatele, které odstraní mezery z názvu skupiny, protože ty podle ISE v názvu skupiny nesmějí být,
- dále se přijatá data naklonují a tím vzniknou dvě skupiny – jedna pro přístup do internetu a druhá pro přístup do lokální sítě. Názvy skupin se obohatí o SSID prefix a o sufix podle přístupu do sítě.
- Dále se vytvoří autorizační profily se jmény totožnými se skupinami a PSK z formuláře. PSK není mapováno na EndpointGroup (tj. je posláno jako prostá hodnota a Symfony se jej nesnaží přidělit instanci EndpointGroup, kterou jinak formulář obsahuje), takže se získá `get` s klíčem „PSK“. Profilům se příslušně nastaví boolean o přístupu do sítě.
- Poté jsou vytvořena autorizační pravidla. Názvy opět mají stejné jako skupiny, `$groupHierarchy` mají nastavené podle paramteru z `ContainerBagInterface` spojený se jménem skupiny.
- Nakonec se `MessageBusInterface` z autowiringu předají zprávy (`ISERequest`) na vytvoření všech objektů na straně ISE serveru. Autorizační pravidla se musí vytvářet jako poslední, protože využívají autorizační profily.
- Po dokončení je uživatel přesměrován na výpis všech skupin (cesta `groups`).

EndpointsController

Controller pro koncová zařízení nabízí plnou CRUD výbavu (tj. Create, Read, Update, Delete). Nejzajímavějšími metodami jsou `edit` a `add`, které upravují, resp. přidávají koncová zařízení. Jelikož mají podobný průběh, bude popsán pro obě zároveň:

- Formulář pro editaci/přidání se liší pouze upravitelností MAC adresy, která nelze po vytvoření zařízení editovat. Formuláři jsou předány potřebné parametry a je zobrazen uživateli.
- Po přijetí žádosti je vyplněný formulář zpracován a je vytvořena instance Endpoint podle požadavků uživatele. Skupina se získává metodou `getTheRightGroup` – izoluje kořen skupiny a podle přístupu k internetu určí správný sufix a zavolá `findOneByRootAndSuffix` repozitáře skupin. Pokud je skupina Unknown, je přístup k internetu nastaven na false.
- Nakonec jsou předány žádosti `MessageBusInterface`, který je předá handlerům. Tím dojde k nastavení firewallu a vytvoření/úpravě zařízení na straně ISE.
- Uživatel je přesměrován na výpis všech koncových zařízení.

HomeController

Zobrazuje pouze statickou stránku s mock obsahem z vývojové fáze. Podle návrhu zde měla být zobrazena aktivní zařízení, ale API ISE neumožňuje získávat informace o statusu připojení. V produkční verzi bude stránka pravděpodobně odstraněna, pokud nedojde ke změně plánu.

ISEController

Jelikož o vytváření a úpravu objektů na straně ISE se starají Controllery příslušné daným entitám, na ISEController zbyla pouze role načítání z ISE. Podle slugu {what} určí, co má načíst (hodnoty policies, endpoints, profiles nebo groups) a přes `MessageBusInterface` pošle žádost o obnovení dat z ISE a jejich uložení do databáze. Po dokončení práce přesměruje uživatele na stránku, ze které na odkaz na obnovení kliknul.

3.2.4 Message a MessageHandler

Třídy ve složce `src/Message` jsou vytvořené podle návrhového vzoru Přepravka a plní funkce předávání zprávy. Po naplnění jejich atributy již nelze

změnit. Momentálně jsou vytvořené třídy pro ovládání firewall (`IPTablesRequest`; obsahuje pouze boolean o povolení přístupu a MAC adresu) a ISE serveru (`ISERequest`; obsahuje příkaz, objekt, o nějž se jedná a typ jako Enum `ISEAPI`).

Třídy ve složce `src/MessageHandler` plní roli zpracovatele zpráv. `IPTablesRequestHandler` se podívá, jestli má přístup povolit, nebo zakázat, zkontroluje, že mu byla opravdu předána MAC adresa a zavolá program s SUID root, který nastaví iptables podle žádosti.

`ISERequestHandler` podle příkazu (jedno z `create`, `refresh`, `update` a `delete`) zavolá obslužnou metodu a předá jí objekt. Obslužná metoda zjistí verzi API, získá z `ISEAPI` správné URL a případně tělo žádosti a odešle HTTP žádost ISE serveru. Návratové hodnoty zatím nejsou testovány. Při provádění příkazu `refresh` nedochází k mazání instancí, které v ISE nejsou.

3.2.5 ISEAPI

Definice API ISE pro potřeby portálu je v Enumu ISEAPI. Obsahuje metody pro vytváření požadovaných URL, metodu pro zjištění používané verze API (protože ISE má verzi ERS a OpenAPI, které zatím nejsou vzájemně zastupitelné) a metody pro vytvoření těla žádosti a vytvoření objektu z odpovědi serveru.

Stejné metody by mohly být i ve vlastních třídách entit, ale uložení do Enumu bylo zvoleno proto, aby byly všechny definice na jednom místě a mohly se jednoduše v případě aktualizací API měnit.

Pozor je třeba dát při vytváření autorizačních pravidel: Cisco má v API chybu a místo slova „Attribute“ místy používá „Attribue“. Další nutností je zahrnutí MAC adresy při úpravách Endpoints, ačkoli to nedává smysl a Endpoint je již jasně určitelný z URL.

Vytváření objektů z odpovědi serveru

K tvorbě instancí dochází následujícím způsobem:

- Na server je odeslána žádost o výpis.
- Server odpoví. `ISERequestHandler` zavolá `ISEAPI::get_response` a vytvoří pole objektů.
- Pole prochází a nad každým objektem volá `ISEAPI::from_array`.
- `ISEAPI` si získá metodou `get_repository` repozitář instancí a pokusí se najít podle ID nebo MAC instanci objektu. Pokud neexistuje, vy-

tvoří ji, pokud existuje, upraví její atributy tak, aby byly v souladu s ISE (metoda `update`).

- Instance je vrácena `ISERequestHandler` a ten ji uloží pomocí `EntityManager`.

3.2.6 Twig

Twig je komponenta Symfony, která umožňuje tvorbu HTML z vlastních šablon. Ty dovolují různé vnořování, cykly, filtry proměnných a tak podobně. Jeho cílem je usnadnit čtení (a tím i tvorbu) šablon tak, aby nebylo třeba používat PHP, které je příliš „upovídané“ (obsahuje zbytečné znaky navíc); jeho syntaxe je zcela minimalistická.

K tvorbě menu je použit balíček `knp_menu bundle`. Tvoří menu pouze z definic typu popisek-odkaz v PHP a umožňuje pokročilé úpravy.

Struktura šablon

Základní šablonou je `templates/base.html.twig`. Obsahuje provázání s používanými knihovnami (FontAwesome, Bootstrap) a přes Webpack Encore include CSS souborů. Dále v sobě má hlavičku stránky spolu s menu a přípravu na modální stránky. Ostatní šablony vkládají svůj obsah do bloku `main`.

Do základní šablony je vložena `templates/nav.html.twig`. Ta zobrazuje postranní menu renderuje jej pomocí `knp_menu_render`. Menu používá CSS třídy z Bootstrap.

Pro každý Controller je vytvořena složka se šablonami, kam se ukládají dle funkce. Ve většině je pouze `index.html.twig`, případně `edit.html.twig` pro úpravy v případě Endpoints a EndpointGroups.

3.2.7 Konfigurace

Konfigurační soubory se nachází ve složce `config`. Hlavním je `services.yaml`, ve kterém se definují služby, které jsou poté využity při autowiringu. Je přidána jedna vlastní služba – `MenuBuilder`, která se stará o vytváření menu v Twigu. V souboru se dále definují parametry aplikace: prefix pro názvy skupin, sufix s významem přístupu k internetu, název skupiny neznámých zařízení, SSID portálu a hierarchie skupin pro autorizační pravidla.

Druhým důležitým konfiguračním souborem je `framework.yaml`. Zde se definují http klienti používaní při volání ISE API. Pod `scoped_clients` jsou definovaní klienti pro ERS API i OpenAPI. Liší se jen URI, oba používají

základní autentikaci HTTP. URI i přístupové jméno a heslo jsou zašifrovány jako proměnné prostředí pomocí komponenty Symfony Secrets.

Posledním důležitým konfiguračním souborem je `doctrine.yaml`. Zde jsou definováni správci entit (entity_managers) pro vlastní databázi projektu i pro databázi Kea DHCP. Také jsou zde konfigurována připojení do těchto databází. Opět jsou uložena jako proměnné prostředí a zašifrována pomocí Secrets.

4 Závěr

Výsledkem práce je funkční Proof of Concept se základní funkcionalitou. Dokazuje, že je systém podle návrhu realizovatelný, že zvolené technologie jsou použitelné a že má smysl se projektu věnovat dále.

Dalšími kroky bude přechod z monolitické architektury na architekturu mikroslužeb, dodělání všech funkcí podle návrhu (zejména propojení s PowerDNS a Kea DHCP a zabezpečení portálu pomocí Firewall komponenty Symfony) a v neposlední řadě zatraktivnění vzhledu aplikace. Poté dojde k uživatelskému testování v provozu ZČU. Po vychytání všech problémů se bude systém používat při výuce.

Seznam obrázků

Seznam tabulek

Seznam zkratek

API Application Programming Interface. 9, 15, 24

DACL Downloadable Access Control List. 10

ISE Identity Services Engine. 5, 8–12, 20, 24

MVC Model-View-Controller. 8, 15, 17

OMZU Obyčejný muž z ulice. 8

ORM object-relational mapper. 15

Literatura

[1] Doctrine. Getting Started with Doctrine - Doctrine Object Relational Mapper (ORM) — doctrine-project.org. `https://www.doctrine-project.org/projects/doctrine-orm/en/2.15/tutorials/getting-started.html`. [Accessed 21-Jun-2023].

[2] Kea. Kea DHCP — isc.org. `https://www.isc.org/kea/`. [Accessed 21-Jun-2023].

[3] PowerDNS. powerdns.com. `https://www.powerdns.com/`. [Accessed 21-Jun-2023].

[4] Yang, X. *Analysis of DBMS: MySQL Vs PostgreSQL*. Kemi-Tornion University of Applied Sciences, 2011.