



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**ZPRACOVÁNÍ A UKLÁDÁNÍ IOT DAT SE
ZAMĚŘENÍM NA LORA SENZOROVÉ SÍTĚ**

PROCESSING AND STORING OF IOT DATA WITH FOCUS ON LORA SENSOR NETWORKS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUcí PRÁCE

SUPERVISOR

TOMÁŠ MORAVČÍK

Ing. JIŘÍ HYNEK, Ph.D.

BRNO 2021

Zadání bakalářské práce



Student: **Moravčík Tomáš**
Program: Informační technologie
Název: **Zpracování a ukládání IoT dat se zaměřením na LoRA senzorové sítě**
Processing and Storing of IoT Data with Focus on LoRA Sensor Networks
Kategorie: Počítačová architektura
Zadání:

1. Prostudujte principy LoRA rádiové technologie se zaměřením na architekturu přenosu dat do cloudu.
2. Analyzujte existující cloudové řešení a standardy Lora Network Servers pro připojení LoRaWAN zařízení. Dále analyzujte způsoby a možnosti jejich integrace s aplikačními servery. Definujte vhodnost použití z pohledu aplikace v chytrých městech, průmyslu a domácnostech jako třech základních business segmentech.
3. Identifikujte vhodný OpenSource LoRa Network server a proveďte jeho nasazení na virtuální server, případně cloudový účet. Potřebný virtuální server nebo cloudový účet zajistí firma Logimic.
4. Navrhněte způsob integrace LoRa Network serveru s aplikačním serverem a proveďte integraci. Přístup k aplikačnímu serveru zajistí firma Logimic.
5. Navrhněte způsob autorizace a identifikaci koncových zařízení LoRa Network serverem a aplikačním serverem s ohledem na minimalizaci úkonů administrátora a navrhněte implementaci.
6. Proveďte praktické testování na sadě alespoň pěti základních LoRaWAN senzorů a proveďte zhodnocení navrženého konceptu. Instalační a konfigurační skripty budou součástí práce. Koncová zařízení poskytne firma Logimic.

Literatura:

- Greengard, S.: *The Internet of Things*. MIT Press, 2015, ISBN 978-026-2527-736.
- Lora Alliance: *Lora Alliance* [online]. 2021 [cit. 2021-10-09]. Dostupné z: <https://lora-alliance.org/>
- SEMTECH, Lora Alliance: *LoRa and LoRaWAN: Technical overview*. 2021 [cit. 2021-10-09]. Dostupné z: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan>
- Lora Alliance: *LoRaWAN. What is it? A technical overview of LoRa and LoRaWAN* [online]. 2021 [cit. 2021-10-09]. Dostupné z: <https://www.tuv.com/content-media-files/master-content/services/products/1555-tuv-rheinland-lora-alliance-certification/tuv-rheinland-lora-alliance-certification-overview-lora-and-lorawan-en.pdf>
- Interní dokumentace firmy Logimic.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Hynek Jiří, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 11. května 2022
Datum schválení: 18. října 2021

Abstrakt

Bakalárska práca sa zaoberá vývojom webovej aplikácie určenej na pridávanie LoRa zariadení do LoRaWAN siete. Aplikácia spĺňa požiadavky progresívnych webových aplikácií, vďaka ktorým je prenosná medzi rôznymi systémami, pričom sa môže správať ako natívna aplikácia. Zariadenia sú pridané na platformu The Thing Stack cloud, ktorá slúži ako LoRaWAN sieťový server. Aplikácia je vyvinutá v jazyku TypeScript v prostredí Angular. Významom práce je zvýšiť efektívnosť a rýchlosť registrovania zariadení do siete a znížiť obtiažnosť procesu. Práca bola vykonaná v spolupráci s firmou Logimic.

Abstract

This Bachelor's thesis deals with developing a web application designed to add LoRa devices to the LoRaWAN network. The application meets the requirements of progressive web applications, which allow the application to be portable and yet to behave like a native application. Devices are added to The Thing Stack cloud platform, which serves as the LoRaWAN network server. The application was developed in TypeScript in the Angular environment. This work aims to increase the efficiency and speed of registering devices to the network and reduce the difficulty of the process. Work was accomplished in cooperation with the company Logimic.

Klíčová slova

Cloud platformy, chytré zariadenia, IoT, LoRA technológia, PWA, webová aplikácia.

Keywords

Cloud platforms, smart devices, IoT, LoRA technology, PWA, web application.

Citace

MORAVČÍK, Tomáš. *Zpracování a ukládání IoT dat se zaměřením na LoRA senzorové sítě*. Brno, 2021. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Hynek, Ph.D.

Zpracování a ukládání IoT dat se zaměřením na LoRA senzorové sítě

Prohlášení

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pán Ing. Jiřího Hyneka, Ph.D. Další informace mi poskytl pán Ing. Michal Valný, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Tomáš Moravčík
17. května 2022

Poděkování

Velmi rád by som sa chcel poďakovať vedúcemu práce Ing. Jiřímu Hynkovi, Ph.D. a pánovi Ing. Michalovi Valnému, Ph.D, za ich ústretovú odbornú pomoc pri riešení zadania práce.

Obsah

1	Úvod	3
2	IoT, LoRa a LoRaWAN	4
2.1	Internet vecí	4
2.2	LoRaWAN	5
2.3	Architektúra	6
2.4	Pripojenie zariadenia do siete	8
2.5	Bezpečnosť	9
2.6	Triedy LoRaWAN zariadení	9
2.7	MQTT	10
3	LoRa Cloud platformy	12
3.1	Cloudové výpočty	12
3.2	Cloudové platformy	13
3.3	Záver	14
4	Progresívne webové aplikácie	16
4.1	Charakteristiky	16
4.2	PWA štatistiky	17
4.3	Manifest	17
4.4	Application Shell	18
4.5	Service Worker	18
5	Analýza problému	19
5.1	The Things Stack cloud	19
5.2	Logimic	20
5.3	Užívateľ	21
5.4	Súčasnú riešenie	22
5.5	Zhrnutie požiadavkov	22
6	Návrh riešenia	24
6.1	Architektúra	24
6.2	Komunikácia	25
6.3	Model toku	25
6.4	Užívateľské rozhranie	27
7	Implementácia	28
7.1	Technológie	28

7.2	Aplikačná logika	29
7.3	Užívateľské rozhranie	31
8	Testovanie	35
8.1	Aplikačné testovanie	35
8.2	Referenčné zariadenie	36
8.3	Návrhy na zlepšenie	37
8.4	Zhrnutie	37
9	Záver	38

Kapitola 1

Úvod

V modernom svete technológie a nové zariadenia čoraz viac zasahujú do bežného života ľudí. Internet vecí (IoT) práve opisuje všetky tieto zariadenia, s prívlastkom chytré, ktoré dokážu po sieti ovplyvňovať okolitý svet bez potreby dozoru užívateľa.

Digitálne technológie prinášajú radikálne zmeny do množstva priemyselných odvetví. Pevné linky, papierové mapy, tradičné kamery začali zanikať, knihy a magazíny taktiež naberajú na poklese. Čím ďalej tak viac zariadení je pripojených na Internet. Mediálny obsah je streamovaný zo serverov, navigačné systémy fungujú na senzoch a satelitoch. Vývoj a produkcia nových, priemyselných strojov – od poľnohospodárskych zariadení po lekárske vybavenie – prenášajú dáta na Internet, kde sú uložené v databázach pripravené na analýzu.

Sila zariadení IoT spočíva v možnosti mať kontrolu i nad detailami prostredia ovplyvňujúceho obrovským množstvom zariadení, ktoré po konfigurácii už nepotrebujú fyzickú obsluhu na správny beh ich činnosti. Napríklad záhradný systém, ktorý sa vie každé ráno zapnúť a popolievať na základe vlhkosti pôdy, zmeny teploty. Alebo činnosti ako kontrola kamier, osvetlenia a uzamknutie domova sú možné bez fyzickej prítomnosti užívateľa, ktorý sa môže nachádzať na druhej strane sveta.

Jednou podskupinou IoT technológií je Long Range (LoRa), určená pre rozsiahle siete s nízkou náročnosťou na energie a malým objemom prenášaných dát zo zariadení. Táto technológia je inovatívna pre odvetvia poľnohospodárstva, zdravotníctva, chytrých miest, logistiky a mnohých ďalších, kde automatizácia zariadení drasticky zvýši efektivitu práce tým, že pracovníci napríklad nemusia osobne kontrolovať stav inventára, ale môžu sa zamerať na podstatnejšie problémy. Zameraním tejto práce je analyzovať momentálne riešenia pripájania koncových zariadení do LoRa siete, poukázať na problematiku procesu a vyvinúť riešenie, ktoré bude jednoduchšie a efektívnejšie než momentálne možnosti. Riešenie bude mať podobu webovej aplikácie, ktorá na základe nových technológií nebude viazaná na jeden typ webového prehliadača alebo systému zariadenia, ale bude prístupná viacerým populárnym voľbám používateľa.

Kapitola 2 sa zaoberá technológiou LoRa, v ktorej sa opisuje architektúra a komunikácia v sieti. Kapitola 3 obsahuje rozbor vybraných cloudových platforiem schopných pracovať s LoRa zariadeniami. V kapitole 4 sa píše o koncepte PWA aplikácií a o výhodách použiteľných v riešení. Kapitola 5 zahrňuje analýzu problému, kde sa predstaví platforma, na ktorú sa riešenie viaže, analýzy potrieb aplikácií a súčasného riešenia. Kapitola 6 predostrie návrh riešenia na základe spomínanej problematiky. Implementácia riešenia a použité technológie sú opísané v kapitole 7. Kapitola 8 sa zameriava na testovanie, v ktorom sa skúma funkcionálnosť riešenia a posledná kapitola 9 prináša zhrnutie práce.

Kapitola 2

IoT, LoRa a LoRaWAN

Táto kapitola oboznámi čitateľa s tematikou a technológiami využitých v tejto práci, objasní terminológiu a vysvetlí koncepty riešenia. Základom je porozumenie IoT odvetvia, ktoré silno naväzuje na LoRa technológie, s ktorými sa práca zaoberá. Hlavný zdroj pri tvorbe tejto kapitoly pochádza z dokumentácie [14] poskytnutej priamo od spoločnosti Semtech, ktorá vyvinula a udržiava LoRa technológiu.

2.1 Internet vecí

Internet vecí reprezentuje elektrické alebo elektronické zariadenia rôznych veľkostí a schopností, ktoré sú pripojené k Internetu [15]. Zariadenia sú schopné identifikácie s inými zariadeniami v sieti a prijímať a posielať dáta. Tie môžu byť prakticky čokoľvek — počítače, telefóny, hodinky, žiarovky, zámky a mnoho iných. Každé z týchto zariadení má svoje unikátne identifikačné číslo (UID) a IP adresu. Môžu byť pripojené káblom, linkou a bezdrôtovo, napríklad celulárne, satelitmi, Wi-Fi alebo cez Bluetooth. Zariadenia dokážu komunikovať vysokofrekvenčnou identifikáciou (RFID) alebo komunikáciou na blízko (NFC) [4].

Veľký význam IoT spočíva v tom, že keď sa zariadenie dokáže reprezentovať digitálne, tak sa stáva súčasťou niečoho väčšieho. Už sa nevzťahuje iba na používateľa ale už aj na prostredie, v ktorom sa nachádza, na pripojené objekty v okolí a na databázu dát. Keď mnoho objektov pracuje v jednote, hovorí sa, že majú „okolitú inteligenciu“ (ambient intelligence).

Podľa *Cisco Systems* [4], ktoré zaviedli Internet of Everything (IoE) Index, biznisy generujú o \$613 miliárd viac profitu ročne ako následok pripojených zariadení, pri čom ale toto reprezentuje len 50 percent toho potenciálu Internetu vecí. Ďalej odhaduje, že hodnota môže dosiahnuť \$14.4 biliónov v profite v nasledujúcej dekáde.

Koncept internetu vecí je náročné jasne definovať, pretože každý fyzický objekt sa môže stať IoT, ak je pripojený k internetu na komunikáciu, kontrolu alebo výmenu informácií. Od chytrých spotrebičov po webové kamery, ktoré možno kontrolovať z chytrého mobilu, sú IoT zariadenia. Objekty ako aj samo-jazdné auto alebo lietadlá sa stávajú IoT zariadeniami, alebo sú minimálne o vylepšené IoT komponentami, ako napríklad súčiastkami na kontrolu funkčnosti motorov. IoT popisuje svet, v ktorom takmer všetko môže byť zapojené a inteligentne spolu komunikovať. Pôvodne stanovený dôraz na M2M aplikácie (stroj so strojom – machine to machine) bol posunutý od výrobcov a biznisov už na dosah všetkých užívateľov. V roku 2021 [7] bolo viac než 10 miliárd aktívnych IoT zariadení a očakáva sa, že do roku 2030 bude počet aktívnych zariadení presahovať 25.4 miliárd. IoT zariadenia dokážu

zbierať informácie z mestských a verejných priestranstiev a ovplyvniť prostredie, verejnú bezpečnosť a manažment zdrojov. Postupom času sa firmy, ale i mestá stanú chytrými a zapojenými do IoT systému, premieňajúc fyzický svet na informačný systém.

Internet of Things možno rozdeliť do 3 hlavných kategórií [2]:

- IoT pre spotrebiteľov – chytré zariadenia do domácnosti ako napríklad žiarovky, televízory, spotrebiče alebo hračky
- IoT pre podniky – chytré merače, bezpečnostné alebo monitorovacie systémy.
- IoR pre priemysel – technológie pre chytré mestá ako monitorovanie dopravy, výrobné automatizácie, predikatívne alebo preventívne zariadenia.

Existujú prvky, ktoré sa môžu prelínať medzi týmito obšírnymi kategóriami, napríklad klimatizácia, osvetlenie a bezpečnosť majú využitie pre spotrebiteľov, podniky a i priemysel.

Bezdrôtové komunikačné protokoly sa rozdeľujú na dve primárne kategórie pre IoT systémy. Sú to protokoly s krátkym a dlhým dosahom. Do prvej skupiny patrí Bluetooth, WiFi a Zigbee. S ich krátkym dosahom sú vhodné na zapojenie v domácnosti. Druhá skupina, s dlhým dosahom komunikácie pre IoT, sú protokoly LoRaWAN a Sigfox. Koncové zariadenia môžu komunikovať v dosahu desiatkach kilometrov, čo ich robí vhodnými na vonkajšie použitie. Táto práca sa zameriava práve na LoRa technológiu s protokolom LoRaWAN.

2.2 LoRaWAN

V roku 2010 vo Francúzsku traja ľudia založili firmu Cycleo s účelom vyvinúť spôsob komunikácie na veľkú vzdialenosť s nízkou cenou energie. Cieľom bolo ovplyvniť priemysel meracích zariadení ich zlepšením. Využili na to Chirp Spread Spectrum (CSS) modulačnú technológiu používanú v sonaroch a radaroch. V 2012 bola spoločnosť prevzatá momentálnym vlastníkom, Semtech. Vtedy taktiež vznikol proprietárny MAC protokol LoRaMAC. Po založení LoRa Alliance® v 2015 sa protokol premenoval na LoRaWAN. Zmyslom LoRa Aliancie je podporiť a propagovať globálne využitie LoRaWAN štandardu [30].

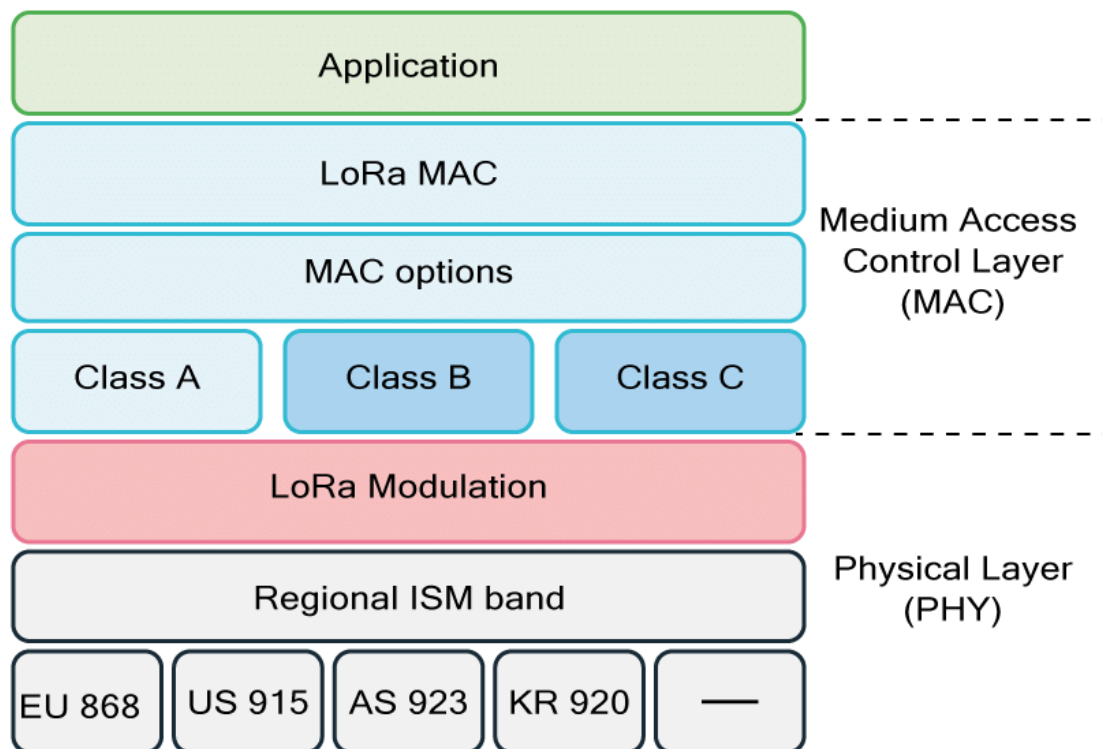
LoRa (Long Range) je technológia rádio-frekvenčnej modulácie umožňujúca nízku mieru prenosu informácií na veľkú vzdialenosť. Presnejšie, je to typ Spread-Spectrum modulácie používajúci „chirp“ (čerp) signál, ktorý sa konštantne mení s frekvenciou. Výhodou tejto metódy je to, že odchýlka v čase a frekvencií odosielateľa a príjemcu bude rovnaká, čím sa výrazne zredukuje komplexnosť príjemcu [11].

Sieť LoRaWAN je typu Low Power Wide Area Network (LPWAN), ktorá zahŕňa zariadenia napojené na batériu a zaručuje obojstrannú komunikáciu. LoRaWAN reprezentuje MAC vrstvu a LoRa fyzickú vrstvu (čip) ???. LoRa sieť je implementovaná hviezdicovou topológiou. Štruktúrou sa môže LoRa architektúra rozdeliť na back-end a front-end, kde back-end reprezentuje LoRaWAN server, ktorý prijíma a ukladá informácie získané zo senzorov. Front-end pozostáva z brán a koncových zariadení. Brány premostujú zariadenia so sieťovým serverom.

LoRaWAN poskytuje rýchlosť prenosu dát medzi 0,3 kbps a 50 kbps. Za účelom maximalizovania životnosti batérie zariadení, zariadenia spravujú svoj prenos dát prostredníctvom Adaptive Data Rate (ADR) mechanizmu. LoRa komunikácia prebieha v nelicencovaných frekvenčných pásmach [10]:

- EU 863-870 MHz pásmo

- EU 433 MHz ISM pásmo
- US 902-928 MHz ISM pásmo
- Čína 470-510 MHz pásmo
- Čína 779-787 MHz ISM pásmo
- Austrália 915-928 MHz ISM pásmo



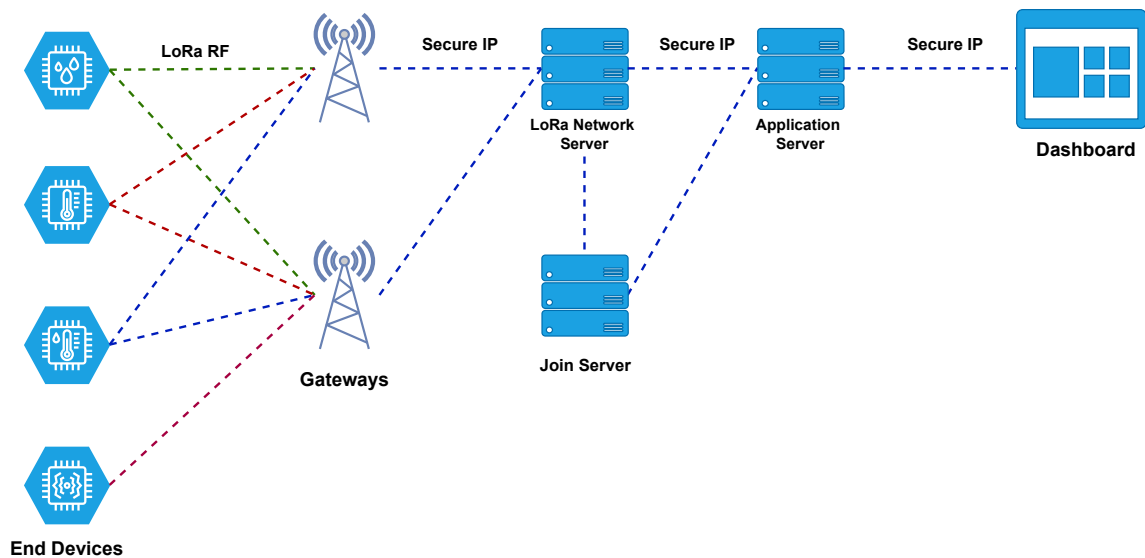
Obrázek 2.1: LoRaWAN Protokol Stack, prevzané z Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed [9].

K hlavným výhodám siete LoRaWAN patrí jej dosah 15 km na vidieku a 5 km v mestských oblastiach. Ďalej, koncové zariadenia na základe prenosu dát vydržia operovať niekoľko rokov na batérii. Kapacitne sieť LoRaWAN podporuje milióny správ. Tento počet závisí nad počtom používaných brán. Podstatným prínosom je taktiež relatívne nízka cena nasadenia a obsluhy. Vďaka svojej architektúre je sieť LoRaWAN ľahko a veľmi škálovateľná s podporou tisícami zariadení a miliónov správ.

2.3 Architektúra

Siete LoRaWAN majú charakter hviezdicovej topológie. Typická sieť LoRaWAN pozostáva z nasledovných prvkov:

- Koncové zariadenia – senzory alebo aktuátory posielajúce/prijímajúce LoRa modulované správy s bránami.
- Brány – posúvajú správy získané zo zariadení na Sieťový Server.



Obrázek 2.2: Architektúra siete LoRaWAN.

- Sieťový Server – LoRaWAN server (LNS), softvér bežiaci na servery, ktorý spravuje sieť.
- Aplikačný Server – softvér bežiaci na servery zodpovedný za bezpečné spracovanie dát.
- Join server – softvér bežiaci na servery, ktorý zabezpečuje vstup zariadení do siete.

2.3.1 Koncové zariadenia

Koncové zariadenia sú senzory alebo aktuátory zapojené do siete LoRaWAN skrze brány prostredníctvom LoRa RF Modulácie. Dokážu získavať informácie z prostredia, digitalizovať ich a poslať ich na sieť alebo prípadne mechanicky reagovať. LoRaWAN zariadenie musí spĺňať tri princípy aby sieť LoRaWAN mohla byť naplno optimalizovaná. Sú to prenosy na veľké vzdialenosti, dlhá životnosť batérie a nízka cena. Príkladom takýchto zariadení sú merače teploty, vlhkosti, energií, vodovodné ventily a mnohé ďalšie. Pri výrobe sú im pridelené unikátne identifikátory, s ktorými sa bezpečne zapoja do siete.

2.3.2 Brány

Brána načúva LoRa modulované RF správy zo všetkých koncových zariadení v prístupnej vzdialenosti a posúva tieto správy na LoRaWAN server. Brána nemá stanové priradenie k zariadeniam, viacero brán môže obsluhovať jedno konkrétne zariadenie v dosahu. Duplikátne správy potom rieši LNS. Ak každé zariadenie pošle desať správ denne, jedna brána dokáže uniesť desaťtisíc zariadení.

Dvojica najpodstatnejších typov prenosu na LoRaWAN serveri sú uplink a downlink správy [13]. Uplink je prenos dát z LoRa koncových zariadení na brány. Downlink je prenos dát v opačnom smere, teda z brán na koncové zariadenia. Z dôvodu jednoduchého LoRa systémového MAC protokolu, ktorý nezahŕňa výber brán, uplink správy je možné prijať všetkými bránami v dosahu. Touto výhodou sa znižuje chybovosti paketov a dovoľuje sa nenáročná geolokácia v prípade, že brány je možné lokalizovať. Na základe stavu kanálov si

LoRaWAN server vyberá optimálnu bránu na downlink komunikáciu, zvyčajne najbližšiu k zariadeniu.

Brány tiež skontrolujú integritu správy, na základe čoho ju posunú na LNS alebo zahodia.

2.3.3 Sieťový Server

Sieťový Server má 3 kategórie funkcionalít: správa operácií, šifrovanie a MAC (Medium Access Control). Do prvej kategórie spadá inštalácia a odinštalácia aplikačných služieb a zariadení. Šifrovanie zahŕňa ochranu dát zozbieraných LoRa terminálmi algoritmom AES-128. Úlohou MAC kategórie je správa zariadení s účelom zlepšiť priepustnosť siete.

Každý LoRaWAN server spĺňa nasledujúce prvky [14]:

- kontrola adresy zariadení,
- potvrdenia o prijatých správach,
- adaptovanie miery prenosu dát podľa ADR. protokolu,
- reagovanie na požiadavky zariadení z MAC vrstvy,
- preposielanie dát uplink správ na patričné Aplikačné Servery,
- radenie downlink správ z Aplikačného Serveru na zariadenie v sieti,
- radenie join-request a join-accept správ medzi zariadeniami a Join Serverom.

2.3.4 Aplikačný Server

Významom Aplikačného Servera je zabezpečiť manipuláciu, riadenie a interpretáciu dát získaných zo senzorov. Uplink správy sa tu dešifrujú a downlink správy sa zašifrujú. Aplikačný Server tiež vytvára obsah downlink správ pre zariadenia v sieti.

2.3.5 Join Server

Join Server zabezpečuje proces pripojenia a aktiváciu koncových zariadení do siete. Signalizuje Sieťovému Serveru spôsob, akým majú byť zariadenia zapojené na Aplikačný Server a definuje ich reláciu. Informácie, ktoré Join Server vyžaduje, sú [14]:

- DevEUI – unikátny identifikátor zariadenia,
- AppKey – aplikačný šifrovací kľúč,
- NwkKey – sieťový šifrovací kľúč,
- identifikátor Aplikačného Servera,
- profil služby koncového zariadenia.

2.4 Pripojenie zariadenia do siete

Špecifikácia LoRaWAN [14] dovoľuje dva typy aktivácie zariadení, preferovane *Over-the-Air Activation* (OTAA) a *Activation by Personalization* (ABP):

1. Charakteristiky OTAA:

- základné parametre sú poskytnuté výrobcami zariadení,
- bezpečnostné kľúče môžu byť pravidelne obnovované,
- zariadenia môžu mať viacero identít v prípade zmeny siete,

- zariadenie je nefalšovateľné.

2. Charakteristiky ABP:

- bezpečnostne slabší proces,
- identifikátory a kľúče sú pri výrobe prispôsobené potrebám,
- zariadenia sú okamžite funkčné a komunikácia s Join Serverom je vynechaná,
- zariadenia sú zaviazané na určitú sieť alebo službu.

2.5 Bezpečnosť

Bezpečnosť siete LoRaWAN sa skladá z dvoch hlavných častí: procedúra *Join* a autentifikácia správ [14]. Procedúra *Join* vytvorí základ pre vzájomnú autentifikáciu medzi sieťou LoRaWAN a koncovým zariadením. Do siete tak môžu vstúpiť len autorizované zariadenia. LoRaWAN MAC a aplikačné správy sú overené na základe ich zdroja, prebieha kontrola integrity a dáta sú počas prenosu zašifrované. Sieť LoRaWAN zabezpečuje, že dáta v prenose neboli pozmenené, zapojené sú len oprávnené zariadenia a že sieť nemožno odpočúvať a zachytávať.

2.6 Triedy LoRaWAN zariadení

Koncové zariadenia LoRa môžu pracovať v jednom z troch módov na základe ich triedy. Všetky zariadenia musia podporovať Triedu A. Trieda B musí podporovať módy z Tried A a Triedy B. Trieda C musí podporovať všetky 3 módy operácii. Módy sa týkajú spôsobu, akým zariadenia komunikujú so sieťou [14].

2.6.1 Trieda A

V tejto triede, koncové zariadenia trávajú väčšinu času v nečinnosti, teda v režime spánku. Po akejkoľvek zmene prostredia, ktoré zariadenie monitoruje, sa zariadenie zobudí a inicializuje uplink s dátami o zmene stavu siete. Následne zariadenie očakáva odpoveď od serveru, zvyčajne po dobu jednej sekundy. Ak nedostane downlink správu počas tohto prijímacieho okna (Rx1), zariadenie sa uspí na krátku dobu a znova sa pokúsi načúvať na odpoveď v druhom okne (Rx2). V prípade žiadnej odpovede, zariadenie sa uspí až dokým nenastane ďalšia zmena stavu. Rozdiel medzi prvým (Rx1) a druhým (Rx2) počúvaním zariadenia sa nastavuje na základe oneskorenia uplink prenosu. V Triede A nie je možné aplikáciou zobudiť zariadenie, preto táto trieda nie je vhodná pre aktuátory.

2.6.2 Trieda B

Nadstavba Triedy A, Trieda B ponúka pravidelne rozvrhnuté, časovo ohraničené príležitosti pre zariadenie získať downlink správy zo siete, na základe ktorých je možno využiť zariadenie aj ako aktuátor. Všetky LoRa zariadenia začínajú módom Triedy A. Na Triedu B je možné prepnúť na aplikačnej vrstve.

Pre správnu funkčnosť komunikácie Triedy B je potrebné zaistiť časovo zosynchronizovaný tzv. *maják* (beacon), ktorý musí byť pravidelne vysielaný sieťou cez brány. Koncové zariadenia musia periodicky prijímať signál z majáku, aby si mohli zrovnať svoj vnútorný čas so sieťou. Tento proces má minimálny vplyv na životnosť batérie a zvyčajne ho stačí

vykonať len pár krát denne. Aby mohla sieť LoRaWAN podporovať Triedu B, tak všetky jej brány musia mať zabudovaný GPS časovací zdroj na presnú synchronizáciu majákov.

2.6.3 Trieda C

Zariadenia Triedy C sú vždy zapnuté a stále načúvajú na downlink správy. Preto ponúkajú najnižšie oneskorenie komunikácie medzi serverom a zariadením. Trieda C taktiež implementuje 2 prijímacie okná, ktoré sa zatovoria len v prípade, že zariadenie posielá prenos na server. Takto môžu získať downlink správú kedykoľvek. Keďže zariadenia v tejto triede majú vysokú spotrebu energie, namiesto batérie sú často napojené káblom. Príkladom takýchto zariadení je pouličné osvetlenie alebo ventily.

2.7 MQTT

V mnohých LoRaWAN implementáciách, je komunikácia medzi bránami a LoRaWAN serverom dosiahnutá prostredníctvom MQTT protokolu. MQTT (Message Queuing Telemetry Transport) je jednoduchý komunikačný protokol určený na posielanie dát medzi zariadeniami. Protokol funguje nad TCP/IP, alebo i na protokoloch, ktoré zabezpečujú usporiadané, bezstratové a obojsmerné spojenia [3]. Bol vyvinutý v 1999 za účelom spoľahlivej komunikácie zariadení v obmedzených oblastiach, s nízkym nákladom na energiu a cenu a pri nižšej šírke pásma. Architektúra je typu Klient-Server a správy sú vzoru publish-subscribe [3]. Protokol zahŕňa tzv. „MQTT Broker“, čo je server, s ktorým klient komunikuje. Väčšinou býva implementovaný na LNS. Klienti nekomunikujú spolu priamo ale cez brokera, ktorý preposiela komunikácie na ostatných klientov. Každý klient môže byť publisher (vysielač), subscriber (prijímač) alebo oboje [12]. MQTT pracuje na základe udalostí a neexistujú periodické dátové prenosi. Vďaka tomu sú prenosi len minimálne.

Správy posielané v prenose sú taktiež minimálne, len s 2 bytovou hlavičkou. Maximálny obsah správy môže byť 256 MB. V MQTT sú správy posielané ako *Topics*, so štruktúrou v hierarchii podľa lomky, napríklad *senzory/teplomer/teplota*. Topics sú vytvorené implicitne a dovoľujú vyberať klientom, o aké správy majú záujem. Server, ktorý bude zbierať, spracovávať a zobrazovať informácie pre užívateľov, bude v rámci MQTT pripojený ako klient – subscriber.

Pre zaistenie spoľahlivosti [6], je možné si vybrať jeden z troch úrovní Quality of Service (QoS).

- QoS 0 – pre minimálny dátový prenos, každá správa je odoslaná raz, bez potvrdenia o prijatí, a správy nie sú uložené pre odpojené zariadenia pre prípad ich znovu pripojenie.
- QoS 1 – broker po odoslaní správy čaká na potvrdenie od klienta. Ak nedorazí v časovom okne, Broker pošle správu znova.
- QoS 2 – klient a broker vykonajú štvor-krokový handshake na zaistenie prijatia správy presne jeden raz.

Pre oba QoS 1 a QoS 2 sú správy uložené pre offline klientov, ktorý nadviazali perzistentnú reláciu.

Aby bol protokol čo najmenší, existuje len päť možných akcií v komunikácii: *publish*, *subscribe*, *unsubscribe*, *ping* a *disconnect*. Význam týchto akcií je:

- Publish posielá správu ako blok dát, ktorý je špecifikovaný pre dané zariadenie alebo implementáciu.

- Subscribe dovolí klientovi prijímať Topic.
- Unsubscribe ukončí príjem Topic správy.
- Ping slúži na kontrolu pripojenia klienta.
- Disconnect posielajú klient serveru za účelom oznámiť ukončenia posielania a prijímania správ. Použitím tejto akcie si klient zaistí rovnakú identity pri opätovnom pripojení.

Bezpečnosť nebola cieľom pri vývoji MQTT, ale za cenu vyššieho dátového zaťaženia možno implementovať SSL/TLS. Momentálne najnovšia verzia je v5.0 vydaná v 2019.

Kapitola 3

LoRa Cloud platformy

Podstatnou úlohou IoT a LoRa systémov je znížiť náklady na nasadenie a údržbu zariadení. Bezdrôtové technológie napomáhajú minimalizovať túto cenu vďaka ich komunikačným protokolom. Nápodobne, cena za obsluhu môže byť znížená využitím cloud technológie, ktoré ponúkajú štandardný framework služieb.

Podľa prieskumu [31], za rok 2019 bolo na svetovom trhu verejne známych 620 IoT platforiem. Výber platformy ovplyvňujú mnohé faktory ako aplikačná doména, protokoly, certifikovaný hardvér, SDK, pripojenie, analýzy a cena.

3.1 Cloudové výpočty

Cloudové výpočty (Cloud computing) sú definované ako dostupnosť zdrojov počítačového systému na požiadanie, hlavne výpočtovej sily a úložiska dát, bez potreby aktívneho riadenia užívateľom [24]. Všeobecne to znamená dostupnosť k dátovým centráram na Internete pre viacerých užívateľov. Cloudové výpočty majú tri základné typy využitia: IaaS, PaaS a SaaS (Interface, Platform, Software – as a Service). IaaS označuje ucelené cloudové služby, PaaS poskytuje rozhranie a softvérové nástroje pre vývojárov a SaaS ako užívateľské aplikácie riadené treťou stranou [26]. V dnešnej dobe sú veľké cloudy distribuované mimo centrálnych serverov, teda pre užívateľa pripojeného na základe vzdialenosti na jeden z týchto distribuovaných serverov, sa nazve okrajový server. Existujú štyri typy cloudov [5]. Prvý typ je verejný cloud, prístupný širokej verejnosti, napríklad cloudy poskytnuté spoločnosťami Google, Amazon a Microsoft. Druhý typ sú privátne cloudy, vyhradené určitej organizácii na privátnej sieti. Nasledujú komunitné cloudy, ktorých infraštruktúra je zdieľaná niekoľkými organizáciami, často so spoločnými záujmami. Štvrtý a posledný typ je hybridný cloud, ktorý je kombináciou predchádzajúcich cloudov, v ktorých sa spravidla rozdeľujú informácie a služby na zverejnené užívateľom a prístupné len pre podnik.

Väčšina IoT cloud platforiem charakterizuje využitie oboch konceptov, LoRaWAN a cloudových výpočtov. Nasledovné IoT cloud platformy vhodné pre komunikačný protokol LoRa budú porovnané na základe ich silných a slabých stránok.

3.2 Cloudové platformy

Existuje mnoho IoT cloudových platforiem, avšak nie všetky podporujú protokol LoRaWAN. V tejto časti priblížime štyri populárne LoRa IoT cloud platformy: The Things Stack¹, Openstack², ThingSpeak³ a Thethings.io⁴.

3.2.1 The Things Stack

IoT cloud platforma The Things Network (TTN) umožňuje energeticky úsporným zariadeniam využiť brány s veľkým dosahom na pripojenie sa do decentralizovanej siete na výmenu dát s aplikáciami. Začiatkom roku 2021 nastala veľká zmena prechodom z infraštruktúry v2 na najnovšiu edíciu v3. Ponechala si naďalej aj svoje bezplatné služby, bez priamej zákazníckej podpory alebo SLA, pod názvom „Community Edition“. Platené plány pre komerčných užívateľov zahŕňujú SaaS (Software as a Service), Enterprise a AWS hosting a zákaznícku podporu. The Things Stack (TTS) nie je závislý na certifikovanom hardvéri a zariadenia sú voľne pripojiteľné. Tento rozbor sa zameriava hlavne na „Community Edition“, ktorého základná funkcionálnosť je prevažne rovnaká ako pri platenej verzii.

Nová, The Things Stack v3 infraštruktúra, priniesla zmeny ako:

- podpora všetkých tried A, B a C a podpora LoRaWAN v1.0.4 a v1.1,
- RX1 Delay,
- príkazy MAC,
- rozšírený formát aplikačných dát (uplink a downlink),
- API rozšírenia.

LoRaWAN v1.0.4 bola vydaná v Októbri 2020 [33] pričom je ale novšia než verzia v1.1 vydaná v Októbri 2017 [32], preto sú zahrnuté obidve špecifikácie.

RX1 Delay je čas za ktorý sa otvorí prvé prijímacie okno. V tomto okne môže LoRaWAN server naplánovať downlink. Druhé okno sa otvorí presne 1 sekundu po prvom. Vďaka RX1 Delay možno toto oneskorenie upraviť medzi 1 a 15 sekundami na prispôbenie sa pri vysokej latencii [18]. TTS očakáva a uprednostňuje LoRaWAN špecifikované príkazy MAC Sieťového Serveru na zariadenia. Ak zariadenie neodpovedá na MAC príkaz, mohla nastať porucha prenosu dát. Tieto nastavenia je ale možné si upraviť v nastaveniach [17].

Správa siete môže byť riadená skrze webové rozhranie alebo cez príkazový riadok (CLI), s ktorým možno využiť skripty.

3.2.2 OpenStack

OpenStack je open-source cloudový manažment systém (CMS), ktorý bol navrhnutý na budovanie IaaS cloud modelov. Tieto modely sú nasadené vo virtuálnych strojoch (VM), schopné riadiť celú platformu. Poskytuje primárne funkcie a rozhrania pre vývojárov aplikačných služieb na Cloude. OpenStack sa skladá z komponentov, ku ktorým si každý môže pridať svoje na základe potrieb. Z toho niektoré komponenty sú považované za kľúčové a udržiavané komunitou, napríklad výpočtový Nova, sieťový Neutron, úložný Swift a mnoho iných. Pozostávajú z kolekcie softvérov založených na jazyku Python, ktoré riadia prístup

¹<https://www.thethingsindustries.com/>

²<https://www.openstack.org/>

³<https://thingspeak.com/>

⁴<https://thethings.io/>

k spoločným úložiskám, výpočtom a sieťových zdrojov. Cloudové nasadenie je poskytované spoločnosťou Red Hat pod názvom Red Hat OpenStack Platform.

3.2.3 ThingSpeak

ThingSpeak je IoT platformová služba určená na analýzu, agregáciu, vizualizáciu dátových prenosov na cloude. S integráciou MATLAB® analytiky je možné vykonať MATLAB kódy na vlastné predspracovanie dát. S bezplatnou možnosťou je vhodný pre študentov, výskumníkov alebo nadšencov IoT, bez potreby nastavovania serverov a webu. Dokáže pracovať aj s platformami ako Arduino, IoBridge, Twitter a ThingHTTP. Platforma má dva kľúčové prvky, kanály (channels) a polia (fields). Kanály obsahujú dátové, lokačné a stavové polia, a polia reprezentujú dáta kanálov, prakticky fungujúce ako premenné. Je možné zbierať dáta na kanály s 8 poliami na spracovanie ľubovoľných dát, 1 pole na stav a 3 na lokalizovanie. Platforma má jednoduchú konfiguráciu a je hardvérovo-agnostická (je použiteľné akékoľvek IoT zariadenie) [19].

3.2.4 Thethings.io

IoT cloud platforma zo Španielska, používaná často na väčšie priemyslové projekty, ponúka back-end pre IoT vývojárov s jednoduchou API. Považuje sa za hardvérovo-agnostický systém a dovoľuje zariadeniam využiť WebSockets, HTTP, CoAP alebo MQTT protokoly. Ďalej poskytuje monitorovanie v reálnom čase, správu zariadení a analýzu.

3.3 Záver

Platformy opísané v sekcii 3.2 majú jednoduché a intuitívne použitie v sférach vývoju aplikácií, správy systému, monitorovania, analýzy a vizualizácie.

ThingSpeak má jednoduché užívateľské rozhranie, bezplatná verzia má ročný limit 3 milióny správ, maximum 4 kanálov a interval medzi jednotlivými správami je minimálne 15 sekúnd. Bezplatný The Things Stack verzia má limit na 10 zariadení, 1 bránu a 1 aplikáciu. OpenStack a Thethings.io neposkytujú žiadne bezplatné hosting služby. Pre nadšencov ale i začiatočníkov tejto technológie sú na malé, nekomerčné projekty najvhodnejšie platformy ThingSpeak a The Things Stack s ich bezplatnou verziou alebo predplátnou verziou vypočítanou na základe počtu zariadení (TTS) alebo počtu správ (ThingSpeak).

Na veľké, priemyselné projekty sú vhodné všetky spomenuté platformy. ThingSpeak využíva aspoň 115 firiem, Thethings.io a Red Hat OpenStack aspoň 16 firiem. The Things Stack tvrdí, že má cez 500 firemných zákazníkov a cez 150 000 užívateľov^{5 6 7 8}.

Pre cloudovú správu na škále chytrých miest je najlepšou voľbou The Things Stack, za ktorým na druhom mieste je ThingSpeak.

The Things Stack je jediná z vybraných platforiem, ktorá má najvyššiu aplikovateľnosť ako LoRa IoT Cloud. Kombináciou všetkých výhod ako jednoduchá konfigurácia zariadení, poskytnuté úložisko IoT dát, analýza, vizualizácia a spravovanie, sa The Things Stack stal najzdatnejšou platformou v porovnaní s inými. Samozrejme, toto hodnotenie nezapiera osobitné výhody ostatných platforiem. Na záver je prehľad platforiem v tabuľke 3.1 s ich technickými rozdielmi: .

⁵<https://enlyft.com/tech/products/red-hat-cloud-suite>

⁶<https://enlyft.com/tech/products/thethings-io>

⁷<https://enlyft.com/tech/products/thingspeak>

⁸<https://www.linkedin.com/company/the-things-industries>

	The Things Stack	RH OpenStack	ThingSpeak	Thethings.io
Protokoly	MQTT, gRPC, HTTP, Websockets, TCP/IP, UDP	MQTT, TCP/IP, Websockets, HTTP	MQTT, HTTP, TCP/IP	MQTT, HTTP, CoAP, Websockets
Hardvér	Agnostický	Red Hat Enterprise Linux (RHEL)	Agnostický	Agnostický
Jazyky	Go, JavaScript	Python	Ruby	Python, Javascript
Cena	Bezplatne alebo na základe plánu, počtu zariadení a nasadenia ⁹	Na základe výberu plánu ¹⁰	Bezplatné nekomerčne, komerčne cena za unit (33 miliónov správ) ročne ¹¹	Na základe počtu zariadení a plánu ¹²

Tabulka 3.1: Charakteristiky vybraných LoRa platforiem

⁹<https://accounts.thethingsindustries.com/fee-calculator>

¹⁰<https://www.redhat.com/en/blog/red-hat-announces-general-availability-of-new-infrastructure-solutions-red-hat-openstack-certification-update>

¹¹<https://thingspeak.com/prices>

¹²<https://thethings.io/new-platform-plans/>

Kapitola 4

Progresívne webové aplikácie

Mobilné aplikácie majú veľký úspech na trhu, a preto rôzne podniky a spoločnosti majú záujem mať svoju vlastnú optimalizovanú aplikáciu. Mobilný užívateľ v priemere strávi až 80% času v nákupných aplikáciách vyvinutých natívne oproti 20% času na webe [1]. V prípade tvorby natívnych aplikácií, cena vývoja rastie vysoko pre jednotlivé platformy. Pokrok vo webových technológiách poskytol nové vlastnosti a rozšírenia, ktoré boli doteraz limitované len pre natívne aplikácie. Toto otvorilo možnosti sústrediť všetok vývoj na webové aplikácie, ktoré bežia vo webových prehliadačoch. S príchodom progresívnych webových aplikácií sa môžu bežné webové stránky javiť a chovať ako nainštalované aplikácie. Prostredníctvom nových konceptov a požiadavok, presadených spoločnosťou Google, sa stali tieto implementácie hodné ich úsilia. Teda PWA je webová stránka implementujúca špecifické požiadavky, ktoré budú ďalej spomenuté. V PWA podporovaných prehliadačoch takáto stránka môže byť pridaná na domovskú obrazovku zariadenia používateľa a byť používaná bez pripojenia na internet. Vyzerá ako bežná aplikácia, avšak beží v upravenom okne prehliadača so skrytými klasickými funkciami jeho rozhrania [25].

4.1 Charakteristiky

Medzi charakteristické prvky PWAs odlišujúce od obyčajných web stránok je pridaná funkcionálna a používateľská skúsenosť. Webové stránky požadujú od užívateľa otvoriť prehliadač, zadať URL a počkať, kým sa obsah stránok stiahne pri každej návšteve, bez možnosti prezerania offline. PWA potrebuje vykonať tieto kroky len pri prvej návšteve. Po inštalácii na domovskú obrazovku, potrebné statické súbory ako HTML, CSS, JavaScript a obrázky webovej stránky budú uložené na mobile užívateľa, pripravené fungovať aj offline. Dynamické dáta môže byť uložené vo vyrovnávacej pamäti a znovu aktualizované napríklad v prípade dostupnosti nových dát alebo keď má mobil primerané internetové pripojenie.

Bežná webová stránka by bola zobrazená v prehliadači (napríklad Chrome pre Android), s viditeľnými prvkami prehliadača ako adresný riadok a menu. PWA beží bez týchto prvkov, čo jej prináša vzhľad bežnej aplikácie. Správne štylizovanú PWA dizajnovanú podľa smerníc, by bolo náročné rozlíšiť od bežnej natívnej aplikácie. V porovnaní PWA s natívnou mobilnou aplikáciou, jednou z hlavných črt je minimálna stopa zanechaná na mobile užívateľa, bez kompromisu na funkcionálnosti.

Z technického hľadiska, progresívna webová aplikácia je webová stránka s meta dátovým JSON dokumentom, s *Application Shell* a skriptom v JavaScripte (*Service Worker*) [23].

Application Shell predstavuje statické GUI (Grafické Uživatelské Rozhranie) a logiku potrebnú na vykreslenie a zobrazenie aplikácie bez dynamického, na pripojení závislom, obsahu. Môže obsahovať smerovanie, navigáciu a UI elementy. Vďaka dostupnosti offline a načítania dát z cache, aplikácia je vykreslená okamžite, čím sa javí ako rýchlejšia aplikácia v porovnaní s webovou.

Service Worker je skript, ktorý riadi všetko sieťové pripojenie, ukladanie logiky do cache a procesy v pozadí, čo umožňuje funkčnosť aplikácie bez pripojenia na internet.

Pre vývojára progresívnych webových aplikácií je nutné viacero zvážiť faktorov. Google sa zastáva názoru optimalizovať čas potrebný na aplikáciu aby bola interaktívna s najväčšou prioritou. Toto sa zakladá na obsiahlom ukladaní dynamického obsahu do vyrovnávacej pamäte, chytřejších žiadostí z webového serveru a načítania obsahu. Google taktiež vydal testovací nástroj pre vývojárov PWAs zvaný *Lighthouse*¹. Nástroj pomáha pri testovaní PWA štandardov webovej stránky a tým uľahčuje jej vývoj.

4.2 PWA štatistiky

Nasledovné informácie poukazujú na výhody a zmysel implementácie PWA, získané z viacerých prieskumov [1, 8].

- Konverzný pomer PWA sa priemerne navýšil o 36%. Tento pomer symbolizuje úspešnosť spracovania návštevníkov webu, čo je pomer počtu stiahnutí voči celkovému počtu zhliadnutí stránky aplikácie. Napríklad, ak 1000 používateľov zhliadlo stránku produktu a 100 z nich si nainštalovalo aplikáciu, konverzný pomer je 10%. Hlavným dôvodom je väčší čas strávený užívateľmi na mobile než na stolnom počítači.
- PWAs majú o 50% viac interakcií s užívateľmi oproti natívnymi aplikáciami. Twitter si s PWA zvýšil počet videných stránok o 65% a Trivago mal angažovanosť užívateľov zvýšenú o 150%.
- Rýchlosť, ktorou sa načíta stránka u PWAs sa znížila 10 násobne. Až 53% užívateľov opustí stránku, ktorá sa načíta viac než 3 sekundy. S vlastnosťami PWA je vyriešenie tejto prekážky kľúčovým aspektom pri tvorbe aplikácie moderných biznisov.
- PWA v porovnaní s natívnou aplikáciou môže stáť v priemere tri až štyri krát menej. Okrem nižšej ceny tieto aplikácie môžu pracovať na rozličných operačných systémoch.
- PWAs sú porovnateľne menšie než natívne aplikácie, zvyčajne pod 1 MB. Twitter Lite dokázal dosiahnuť veľkosť len 193 kB oproti 106.5 MB pre aplikáciu na Android.

4.3 Manifest

Manifest je súbor vo formáte JSON, ktorý predstaví prehliadaču progresívnu webovú aplikáciu a definuje, ako by sa mala správať pri inštalácii na užívateľskom zariadení [20]. Manifest obsahuje typicky názov aplikácie, jej ikonu a URL, ktorá sa má otvoriť pri štarte aplikácie. Prehliadače Chrome, Firefox, Edge, UC prehliadač, Opera a Samsung prehliadač podporujú Manifest súbor, pričom Safari podporuje len čiastočne. Súbor, zvyčajne zvaný `manifest.json` je uložený v koreňovom priečinku stránky. Kľúčovými prvkami manifestu sú [20]:

- `name` – názov aplikácie pod ktorým bude nainštalovaná,

¹<https://developers.google.com/web/tools/lighthouse>

- `icons` – ikony zobrazené po nainštalovaní. Sú uložené ako objekt obsahujúci zdroj, veľkosť a typ obrázku,
- `start_url` – požadované aby prehliadač mohol vedieť, čo mal otvoriť a zamedzuje otvorenia na predchádzajúcej stránke navštívenej používateľom,
- `background_color` – použitá na definovanie farby pozadia aplikácie, využívaná najmä počas načítania aplikácie pre prirodzený prechod od spustenia po načítanie obsahu aplikácie,
- `description` – opisuje zmysel aplikácie,
- mnohé ďalšie vlastnosti ako `display`, `shortcuts`, `scope` a podobne.

4.4 Application Shell

Application Shell obsahuje lokálne zdroje (zvyčajne HTML, CSS a JavaScript) potrebné pre načítanie kostry aplikácie používateľského rozhrania tak, aby fungovala offline a mohla byť vyplnená obsahom s využitím Javascriptu [21]. Obsah stránky je dynamicky načítaný počas navigovania užívateľa aplikáciou. Ak bola *app shell* uložená do vyrovnávacej pamäte skrze service worker, tak pri opakovanej návšteve sa zmysluplné pixely na displeji načítajú o mnoho rýchlejšie bez pripojenia siete. Využitie *app shell* nie je nutným požiadavkom PWAs ale pri správnom použití môže výrazne zvýšiť výkon aplikácie.

4.5 Service Worker

Service Worker je súbor obsah Javascript , ktorý beží separátne od hlavného vlákna prehliadača, zachytáva sieťové žiadosti, ukladá alebo čerpá zdroje z vyrovnávacej pamäte a dodáva push notifikácie [22]. Keďže je *service worker* navrhnutý asynchrónne, lokálna pamäť a synchrónne XHR (*XMLHttpRequest* je používaný na stiahnutie dát bez potreby obnoviť celú stránku) nemožno využiť. Na komunikáciu so stránkou musí použiť `postMessage()` metódu na poslanie dát. Od servera môže dostať push správy i v prípade keď nie je aplikácia aktívna, vďaka čomu používateľ vidieť push notifikácie bez potreby otvorenia prehliadača. Service worker je programovateľná sieťová proxy umožňujúca kontrolu nad sieťovými požiadavkami a ich prispôbením. Preto musí bežať len nad HTTPS kvôli ochrane pred útokmi typu *man-in-the-middle* (odpočúvanie komunikácie medzi účastníkmi) [22]. V prípade potreby uloženia stavov alebo informácií i po reštartovaní aplikácie, je možné využiť *IndexedDB* databázu. *Service worker* sa zakladá na 2 APIs aby aplikácia mohla pracovať offline: *Fetch* príkaz na získanie dát zo siete a *Cache* ako perzistentné úložisko aplikačných dát. Toto *Cache* je nezávislé od Cache prehliadača alebo stavu siete.

Činný Service Worker pozostáva z troch krokov:

1. registrácia,
2. inštalácia,
3. aktivácia.

Registračný krok hovorí prehliadaču, kde sa *service worker* nachádza. Ak prehliadač pokladá *service worker* ako nový, tak nastane pokus o jeho inštaláciu. Nakoniec, v aktivačnom kroku sa aktualizuje cache.

Kapitola 5

Analýza problému

Táto časť práce sa snaží opísať problematiku pridávania zariadení na server LoRaWAN a zamyslieť sa nad potrebami užívateľa. Vybraná platforma, s ktorou riešenie bude komunikovať, je The Things Stack cloud. Okrem kvalitných možností pre bezplatných užívateľov túto platformu taktiež využíva firma Logimic, ktorá toto riešenie bude využívať.

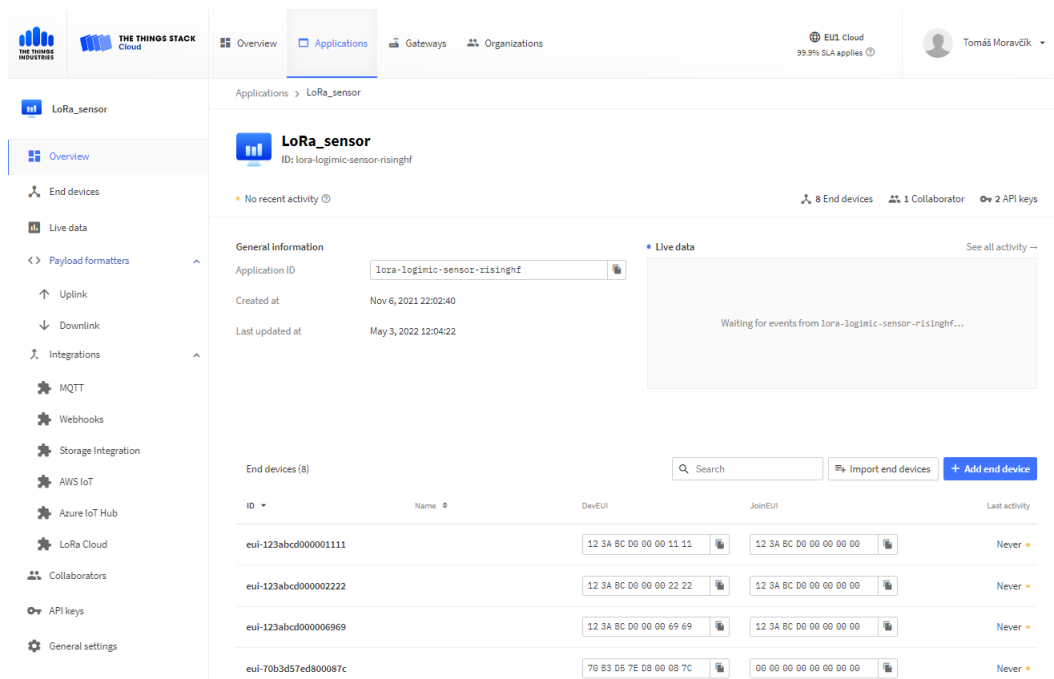
5.1 The Things Stack cloud

Rolu LoRaWAN serveru v tejto práci vyplňa The Things Stack (TTS) Cloud. Každý používateľ má svoju vlastnú izolovanú sieť pričom základná infraštruktúra je zdieľaná s ostatnými zákazníkmi. TTS poskytuje možnosť spravovania zariadení, brán a dát. Je rozšíriteľný, bezpečný a má veľkú aktívnu komunitu. Podporuje všetky LoRaWAN triedy, predchádzajúce LoRaWAN verzie a všetky regionálne parametre definované LoRa Alianciou. Podstatným plusom je možnosť využitia APIs. Prístup na server je možný cez konzolu po vybratí (ideálne) geograficky najbližšieho zhluku. Spoločnosť Logimic na svoju prácu využíva TTS, čo umožnilo vytvoriť si predstavu o škále a potrebách tejto práce. S bezplatne vytvoreným účtom som sa oboznámil s vlastnosťami a funkciami TTS.

Na obrázku 5.1 je menu po ľavej strane, ktoré obsahuje všetky primárne služby. V *Payload Formatters* spracovávajú prichádzajúce a odchádzajúce dáta na koncových zariadeniach. Významom je konvertovať binárne dáta na čitateľné informácie. *Integrations* pomáhajú s manipulovaním dát a vykonávaním akcií na základe spúšťaním udalostí. Príkladom je načasovanie príkazov poslaným zariadeniam.

Po získaní DevEUI z koncového zariadenia je možné kód spracovať a vytvoriť potrebné dáta na pridanie zariadenia do rozhrania TTS cloudu. Aby sa zabránilo situáciám, v ktorej sa naskenuje invalidný kód, je potrebné skontrolovať formát kódu, či vyhovuje regulárnemu výrazu kódu daného zariadenia. TTS Cloud dokáže toto vyhodnotiť a odpovedať správou o neúspešnom pridaní, avšak s jednoduchou kontrolou sa dá predísť potrebe očakávať chybné odpovede.

Po validácii kódu je nutné vytvoriť objekt pridávaného zariadenia vo formáte JSON podľa šablóny TTS cloudu. Koncové zariadenia sú pridané do viacerých registrov. Identity Server obsahuje register s metadátami koncového zariadenia, Network Server register obsahuje MAC konfiguráciu, MAC stav a sieťové relačné kľúče. Application Server ukladá formát časti prenášaných dát (*payload*) a aplikačné relačné kľúče. Join Server ukladá koreňové kľúče (*root keys*) [29].



Obrázek 5.1: TTS Cloud uživatelské rozhranie aplikácie „LoRa_sensor“.

Pri registrácii zariadenia sa každému serveru sa pošle jemu špecifická dátová informácia v odporúčanom poradí:

1. Identity Server (IS)
2. Join Server (JS)
3. Network Server (NS)
4. Application Server (AS)

Pri odstránení zariadení je doporučené ich odstrániť v opačnom poradí. [29]

5.2 Logimic¹

Hlavným produktom firmy je vývoj softvéru na objednávku. Zameriava sa na IoT aplikácie pre obce a priemysel. Klientela firmy Logimic je obširna. Pochádzajú prevažne z európskych krajín a US. Logimic sa snaží čo najviac pripraviť riešenie pre ich nový systém, napríklad vyvinutie rozhrania a zaistenie hardvéru. Pri nasadzovaní služieb potrebujú pripraviť veľké množstvá koncových zariadení, čo významne ovplyvňuje čas doby inštalácie a sprevádzkovania systému. Okrem dlhého procesu registrovania zariadení je taktiež potrebné oboznámiť klienta o spôsobe pridania týchto zariadení do siete, pretože poloha klienta často nie je prístupná personálu firmy Logimic na osobné stretnutie. Logimic teda potrebuje riešenie, ktoré bude jednoduché, intuitívne, ľahko prenosné a efektívne na prácu s koncovými zariadeniami.

¹<https://www.logimic.com/>

Register end device

From The LoRaWAN Device Repository [Manually](#)

Frequency plan ⓘ *

Select... | v

LoRaWAN version ⓘ *

Select... | v

Regional Parameters version ⓘ *

Select... | v

[Show advanced activation, LoRaWAN class and cluster settings](#) v

DevEUI ⓘ *

. Generate 4/50 used

AppEUI ⓘ *

. Fill with zeros

AppKey ⓘ *

. Generate

End device ID ⓘ *

my-new-device

This value is automatically prefilled using the DevEUI

After registration

- View registered end device
- Register another end device of this type

Register end device

Obrázek 5.2: TTS Cloud rozhranie aplikácie pre zapojenie zariadenia do siete.

5.3 Užívateľ

So širokou škálou zákazníkov v rôznych odvetviach nie je možné očakávať, že užívateľ bude technicky zdatný v tejto problematike. Je potrebné uvažovať o užívateľovi ako o laikovi a tak prispôbiť riešenie. Aplikácia preto nemôže byť príliš komplexná ale naopak, intuitívna. Užívateľ taktiež môže presunúť prácu priradovania zariadení svojim zamestnancom, alebo niekomu inému. Preto by aplikácia taktiež nemala vyžadovať administratívnu znalosť systému. Keďže klientom, respektíve používateľom, môže byť každý, riešenie by malo ľahko rozširiteľné a nezávislé na hardvéri užívateľa.

5.4 Súčasné riešenie

K najdôležitejším funkciám patrí registrácia zariadení do siete, ktoré sa manuálne pridávajú v rozhraní TTS cloudu, ako možno vidieť na obrázku 5.2. Zariadenia sú spravované pod aplikáciami. Jedna aplikácia môže mať neobmedzený (podľa kúpeného plánu) počet zariadení. Zariadenia sa pridávajú prostredníctvom konzoly alebo terminálu *CLI*.

5.4.1 Konzola

Konzola referuje k TTS cloud webovému rozhraniu 5.1. Po kliknutí na *Add end device* sa dostaneme na 5.2 stránku. Odtiaľto možno vyhľadať zariadenie v databáze TTS cloudu, čo napomôže vyplniť niektoré parametre. V prípade manuálneho pridania, treba vyplniť niekoľko rôznych parametrov [27]:

- *Frequency plan* – príslušné k regiónu nasadenia, zariadenia a brány musia mať rovnaký frekvenčný plán na komunikáciu,
- *LoRaWAN version* – táto informácia je poskytnutá so zariadením pod názvami *LoRaWAN version*, *LoRaWAN specification* alebo *MAC version*,
- *Regional Parameters version* – poskytnuté výrobcom zariadenia,
- *DevEUI* – 64 bitový unikátny identifikátor poskytnutý výrobcom zariadenia,
- *AppEUI* – 64 bitový unikátny identifikátor, môže byť poskytnutý výrobcom alebo vygenerovaný, taktiež referovaný ako *JoinEUI*,
- *AppKey* – pre zariadenia LoRaWAN verzie 1.0.x, môže byť poskytnuté výrobcom alebo vygenerovaný,
- *NwkKey* – pre zariadenia LoRaWAN verzie 1.1.x, môže byť poskytnuté výrobcom alebo vygenerovaný,
- *End device ID* – unikátne ID vybrané užívateľom.

Výber nesprávnej LoRaWAN verzie môže viesť k zložitým problémom, ktoré narušia konzistenciu komunikácie zariadenia. Pri rozšírených nastaveniach možno vybrať typ aktivácie na OTAA alebo ABP a výber triedy zariadenia (A, B, C). Taktiež tu možno nastaviť Rx2 mieru prenosu dát a frekvenciu. TTS cloud má jeden spôsob pridania viacerých zariadení na server nahraním súboru vo formáte JSON, v ktorom sú informácie o zariadeniach definované podľa TTS referencie.

5.4.2 CLI terminál

Existujú dve verzie TTS CLI: `ttn-lw-cli` a `ttn-lw-cli`. Prvá verzia je dostatočná pre väčšinu používateľov a zahŕňa všetky príkazy na správu TTS. `ttn-lw-cli` obsahuje dodatočné príkazy pre správu viacerých užívateľov (*tenants*) na spoločnom TTS účte pre platený plán *Enterprise*. Po inštalácii klienta možno poskytnutými príkazmi riadiť TTS účet [16].

5.5 Zhrnutie požiadavkov

Pri poskytovaní sietí LoRaWAN na platforme TTS cloud, Logimic často potrebuje spolupracovať s klientmi pre zavedenie zariadení do sietí u nich lokálne, pričom klient musí mať aspoň základné znalosti platformy TTS na pridanie zariadenia. Proces pridávania koncových zariadení na TTS cloud je zdĺhavý a nepraktický. V bežnom prípade bude skupina

rovnakých zariadení spadať pod svoju vlastnú aplikáciu. Zariadenia pod jednou aplikáciou budú mať všetky parametre identické okrem troch. Okrem týchto znalostí klient potrebuje venovať značný čas tomuto úkonu, keďže je nutné registrovať zariadenia osobitne a manuálne. Na základe veľkosti siete tiež klient môže prenechať niektoré časti práce svojim zamestnancom alebo iným laikom, ktorí môžu ľahko vyvolať chybu pri výbere potrebných parametrov zariadenia pri jeho registrácii do siete. Riešenie musí byť dostatočne jednoduché, intuitívne a rýchle, aby ho mohol využiť bežný používateľ bez potreby poznania, ako funguje sieť LoRaWAN alebo TTS cloud platforma. Zariadenia by mali byť vyplnené na základe šablóny, ktorá vyžaduje len DevEUI zariadenia. Ďalším významným faktorom sú rôzne zariadenia a systémy. Klienti môžu používať *macOS* zariadenia a iní môžu *Windows* systémy. Preto by riešenie nemalo byť viazané na špecifický hardvér ale malo by byť ľahko rozšíriteľné.

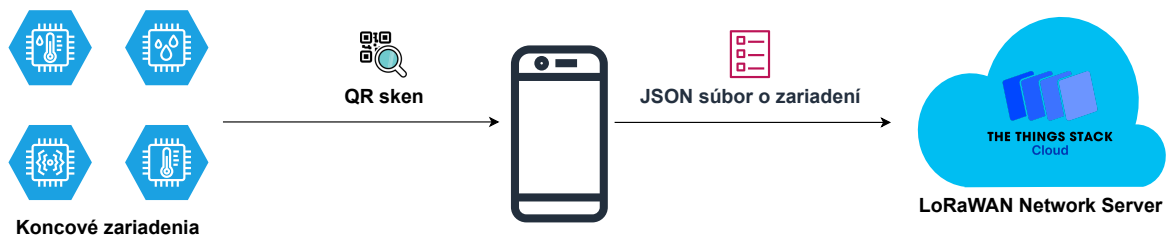
Kapitola 6

Návrh riešenia

Návrhom riešenia tejto práce je prenosná webová aplikácia, ktorá s využitím The Things Stack cloud APIs komunikuje s LoRaWAN serverom. Tým sa výrazne znižuje potreba užívateľa poznať prostredie The Things Stack (TTS) a nutnosť vlastniť dedikovaný hardvér. Webová aplikácia poskytne jednoduchý prístup k funkciám na skenovanie a registrovanie zariadení do TTS cloudu.

6.1 Architektúra

Aplikácia spustená na zariadení užívateľa bude prostredníctvom kamery skenovať QR kódy koncových zariadení, spracuje tieto informácie a pošle ich cez internetové pripojenie na TTS cloud prostredníctvom *HTTP* metód. Architektúra 6.1 je prenosná medzi viacerými systémami ako napríklad *macOS* a *Android*.



Obrázek 6.1: Architektúra aplikácie.

Prihlásením sa užívateľ v aplikácii dostane na stránku s výberom koncového zariadenia, ktoré bude pridávať do siete. Výber zariadenia taktiež implicitne vyberie LoRa aplikáciu, do ktorej sa zariadenia prihlásia. Je to z toho dôvodu, že rovnaké zariadenia budú spadať spravidla práve pod jednu aplikáciu. Po vybratí zariadenia užívateľ môže začať skenovať zariadenia prostredníctvom kamery na mobilnom telefóne, prípadne web kamerou na počítači. V prípade poškodeného QR kódu alebo z iného dôvodu kedy užívateľ nie je schopný použiť kameru, je možné zadať kód do aplikácie manuálne, pretože zariadenie zahŕňa i kód v čitateľnej podobe.

6.2 Komunikácia

Komunikácia s TTS prebieha cez internetové pripojenie aplikácie. TTS poskytuje referenčnú dokumentáciu k API, presnejšie *gRPC* a *HTTP APIs*. V tomto prípade sa využije *HTTP APIs*. Na začiatku je potrebné vygenerovať API kľúč, s ktorým bude možné autorizovať požiadavky na cloud. Kľúč sa získa v rámci užívateľského rozhrania TTS Cloud. API kľúč a OAuth prístupový token majú nasledovný formát:

```
<token-type>.<token-id>.<token-secret>
```

token-type definuje, či sa jedná o API kľúč (NNSXS) alebo OAuth prístupový token (MFRWG). *token-secret* je nutné uchovať v tajnosti, kvôli bezpečnosti pred vniknutím. Pri autorizovaní žiadosti je potrebné použiť celý tvar tokenu. Rozsah API kľúčov určuje entita, z ktorej boli vygenerované [28]. Môžu byť:

- užívateľský API kľúč,
- aplikačný API kľúč,
- API kľúč brány,
- API kľúč organizácie.

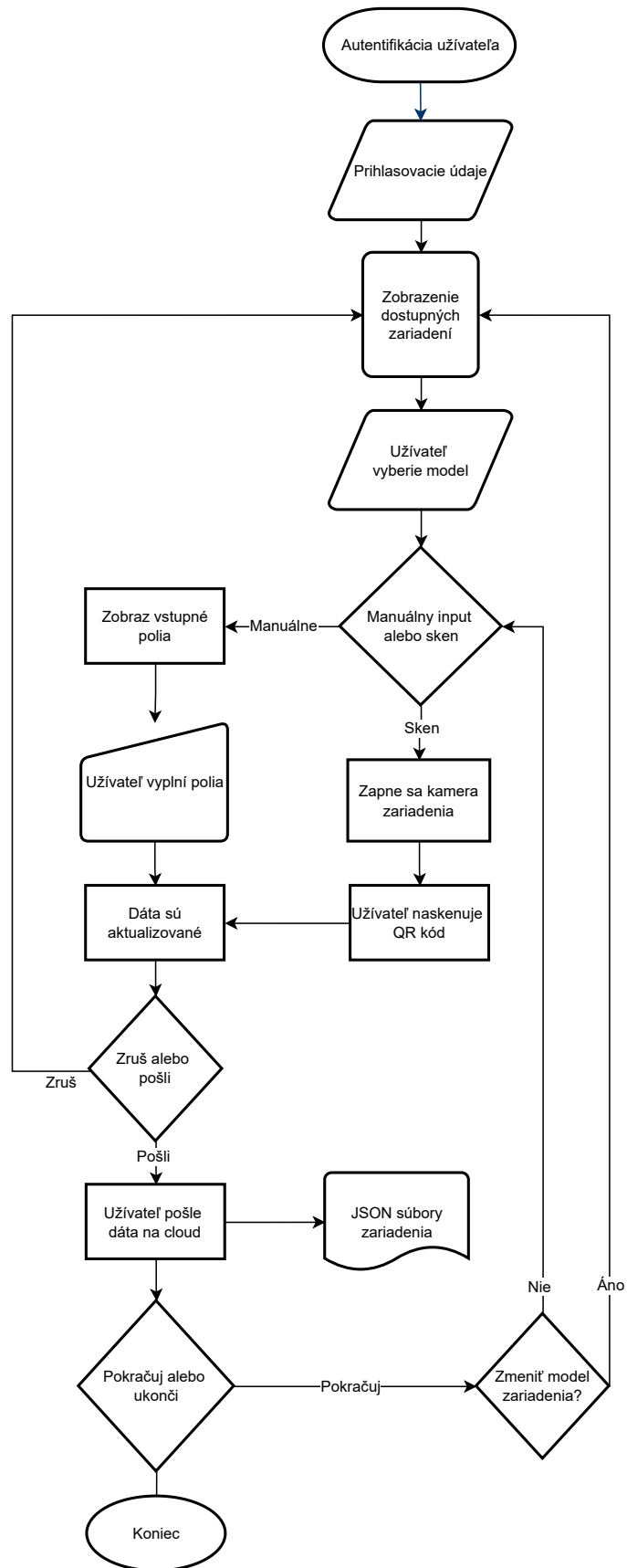
Užívateľský alebo Aplikačný API kľúč pre potreby aplikácie spĺňa nutné prístupové práva. Tento kľúč je priradený užívateľovi prostredníctvom firmy Logimic a v aplikácii by nemal byť dosiahnuteľný ani meniteľný. API kľúč umožní aplikácii vykonávať metódy *GET*, *POST* a *PUT*, s ktorými je možné pridať zariadenie do siete.

Vytvorené súbory JSON sa pošlú serverom TTS metódami *POST* a *PUT*. Následne môže aplikácia očakávať úspešnú odpoveď v podobe novo vytvoreného zariadenia alebo odpoveď o zlyhaní pridania zariadenia.

Metóda *POST* je využitá na zmenu a aktualizáciu objektu. Dáta sa posielajú sa Identity Server a metóda *PUT* sa používa na vytvorenie alebo prepísanie objektu. Táto žiadosť sa zavolá pre Name Server, Join Server a Application Server.

6.3 Model toku

Užívateľ sa dostane po otvorení aplikácie na stránku s prihlásením, tak ako je to naznačené na modeli 6.2. Následne si vyberie zariadenie, ktoré chce oskenovať, a dostáva sa do pracovnej slučky. Tá pozostáva zo skenovania a registrovania zariadení alebo písania kódov manuálne. Užívateľ počas celého procesu môže ľubovoľne meniť aspekty ako výber zariadenia alebo spôsob získania kódu.



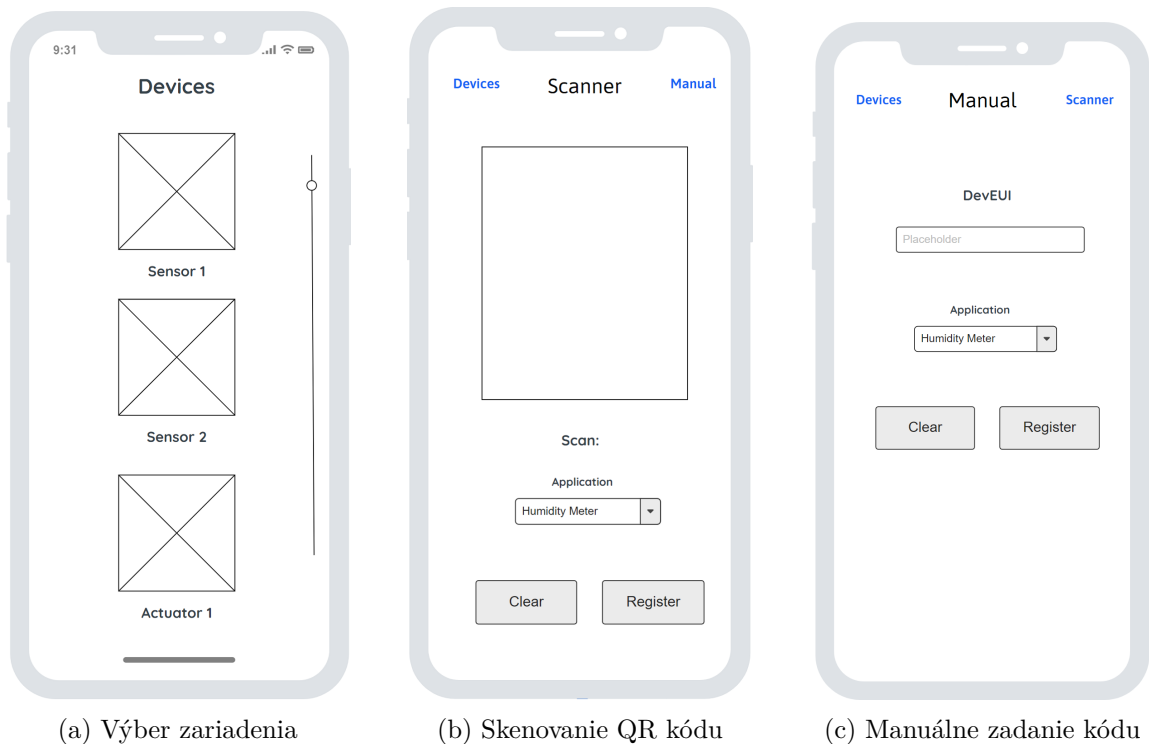
Obrázek 6.2: Proces užívateľskej aktivity a spracovania dát v aplikácii.

6.4 Uživatelské rozhranie

Rozhranie aplikácie by malo byť intuitívne a užívateľ by sa nemal cítiť stratený v jednotlivých stránkach. Po prihlásení so svojimi údajmi, užívateľ bude mať na výber zariadenia 6.3a, ktoré sa chystá pridať do siete. Jednotlivé zariadenia obsahujú obrázok zariadenia s jeho názvom alebo popisom. Výber zariadenia implicitne ovplyvní výber TTS cloud aplikácie.

Po výbere sa užívateľovi zobrazí stránka so skenerom 6.3b, ktorá otvorí kameru alebo ho vyzve o povolenie použiť kameru zariadenia. Snímanie sa zobrazí v naznačenom štvoruholníku a po zacytení QR kódu sa zobrazí notifikácia o úspešnom skene. Užívateľ vidí výsledok skenu v časti *Scan:*. Vo výberovom poli má možnosť zmeniť aplikáciu, pod ktorú zariadenie bude spadať, a stlačením tlačidla *Register* vygeneruje a odošle potrebné dáta na TTS cloud. Notifikácia informuje užívateľa o úspešnom alebo neúspešnom pridaní zariadenia na TTS cloud. Naskenovaním nového zariadenia sa dáta aktualizujú alebo s tlačidlom *Clear* sa zresetujú do pôvodného stavu.

Užívateľ vždy môže prejsť späť na stránku *Zariadenia* 6.3a alebo prejsť na stránku *Manual* 6.3c, ktorá mu dovoľí ručné zadanie kódu. Využíva rovnaké funkcie ako stránka *Scanner* 6.3b, len s rozdielom zmeny snímania kamery na užívateľský vstup.



Obrázek 6.3: Návrh grafického rozhrania

Kapitola 7

Implementácia

Implementácia aplikácie s názvom *pwa scanner* je webová aplikácia nezávislá na platforme. Je funkčná pre prehliadače ako Chrome a Safari. Bola snaha dodržať vývojový diagram 6.2 tak, aby riešenie bolo prehľadné a intuitívne. Kapitola ďalej opisuje jednotlivé celky a funkcionálnosť aplikácie.

7.1 Technológie

Aplikácia je prevažne napísaná v jazyku TypeScript (silne typovaný jazyk rozširujúci JavaScript) vo webovom frameworku Angular. Angular je zameraný na vývoj moderných webových aplikácií. S využitím konceptu PWA je aplikácia ľahko prenosná. PWA je podporovaná na systémoch ako iOS, Android, Windows a Mac, vďaka čomu aplikácia prístupná mnohým užívateľom.

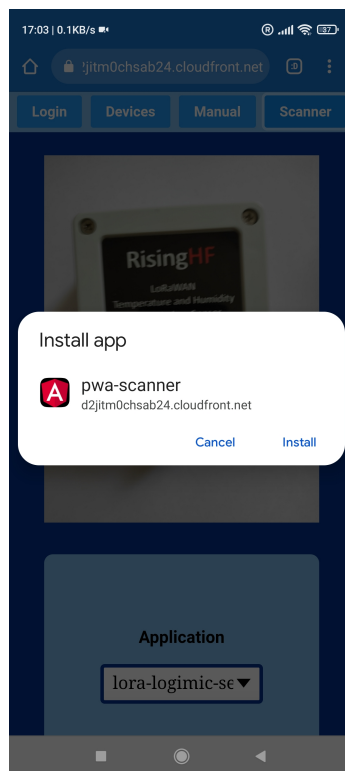
Významné knižnice použité v aplikácii sú *PrimeNG* pre štylizované komponenty, *angular2-notifications* pre informovanie užívateľa a *ngx-scanner*, ktorá umožnila prácu s kamerou a zaznamenávanie QR kódov. PrimeNG obsahuje množinu open-source natívnych komponentov pre Angular. Niektoré z využitých PrimeNG komponent sú *button*, *dropdown*, *message* a *dialog*. Predstavujú predpripravenú štylizovanú jednotku užívateľského rozhrania, ako sú napríklad tlačidlá použité v riešení. *angular2-notifications* je knižnica na prácu s notifikáciami, ktorá sa skladá z komponentov a služieb. Komponenta `<simple-notifications/>` sa vloží na miesto v aplikácii, kde voláme upozornenia a so službou `NotificationsService` riadime logiku notifikácie. V riešení sú využité pri skenovaní a registrovaní zariadení.

Angular taktiež podporuje vývoj PWA, čo značne uľahčilo integráciu technológie. Dôležitým prvkom pri výbere tohto prostredia bol aj podnet od firmy Logimic, ktorá využíva Angular, a tak by aplikácia nevychádzala zo zaužívanej normy firmy a mohla byť v budúcnosti ľahko rozšíriteľná. Progresívne webové aplikácie poskytujú možnosť inštalácie aplikácie na zariadenie užívateľa. Súbor `manifest.webmanifest` obsahuje názov, URL, tému pozadia a ikony aplikácie. Tieto informácie použije súbor service worker `ngsw-worker.js` pri inštalácii aplikácie. Súbor service worker je dynamicky vytvorený prostredníctvom modulu `ServiceWorkerModule`, ktorý sa nachádza a zavolá v súbore `app.module.ts`. Tento súbor zahŕňa všetky moduly použité v aplikácii.

Pri prvom otvorení aplikácie vo webovom prehliadači je užívateľ oboznámený s možnosťou inštalovať (obrázok 7.1) a pridať aplikáciu na domovskú stránku svojho zariadenia. Vytvorí sa ikona aplikácie na domovskej stránke a užívateľ cez ňu môže aplikáciu otvoriť a používať, ako keby bola natívna aplikácia.



(a) Ponuka inštalácie



(b) Potvrdenie inštalácie

Obrázek 7.1: Inštalácia aplikácie

7.2 Aplikačná logika

Aplikačnú logiku predstavujú hlavne služby, ktoré komunikujú s komponentami a The Thing Stack cloudom.

7.2.1 Dáta

Táto časť popisuje cestu dát zariadenia v aplikácii. Aplikácia prijíma dáta o zariadení snímaním kamery alebo užívateľským vstupom. V aplikácii tieto dáta predstavujú reťazce, uložené v komponente *scanner* alebo *manual*, na základe činnosti užívateľa. Ďalej je potrebné vytvoriť z tohto reťazcu objekt. Objekt je inštancia, ktorá obsahuje množinu párov *klúčov a hodnôt*. Na jeho vytvorenie sa využívajú funkcie súboru `RHF1S001.ts`. Tento súbor obsahuje okrem funkcií i pripravené „šablóny“ objektov, ktoré sú špecifické pre TTS cloud. Máme štyri rôzne objekty, jeden pre každý TTS cloud server 5.1 (IS, JS, NS a AS). Objekty majú rozličné štruktúry, ale všetky potrebujú vyplniť trojicu identifikátorov zariadenia. Tie sú DevEUI, AppEUI (JoinEUI) a Device ID. Objekt pre Join Server ako jediný vyžaduje navyše AppKey. Súbor `RHF1S001.ts` obsahuje funkcie, ktorých návratová hodnota je práve objekt pre daný server. Pred poslaním objektov na TTS cloud, je najprv nutné ich previesť na reťazce formátu JSON. Odoslanie dát na TTS cloud je posledný bod manipulovania s dátami zariadenia v aplikácii.

7.2.2 Služby

Veľkou podstatou aplikácie a Angular frameworku sú služby (services). Činnosť služieb je distribuovať dáta a komunikáciu medzi komponentami a prijímať dáta zo serverov. Komponenty môžu využívať služby prostredníctvom tzv. vkladaním závislostí (*dependency injection*), čo je koncept vloženia objektov do iného objektu, na ktorých je závislý.

Aplikácia používa štyri služby, z ktorých najpodstatnejšia je `the-things-stack.service.ts`. Obsahuje funkcie na komunikáciu s TTS cloud servermi prostredníctvom knižnice HTTP z `'@angular/common/http'`. Príkladom je funkcia, ktorá sa dotazuje s dátami na Identity Server TTS cloudu:

```
regIS(data:object, appName:string): Observable<any>{}
```

Parameter `data` typu objekt obsahuje dáta o zariadení upravené pre Identity Server a `appName` je reťazec, ktorý identifikuje aplikáciu na TTS cloudu, pod ktorú zariadenie bude spadať. `Observable<any>` predstavuje asynchrónnu odpoveď od TTS cloudu v podobe reťazcu formátu JSON. Význam zvyšných služieb ako `ui-devs`, `ui-manual` a `ui-service` je zobrazovať a skrývať komponenty, aby aplikácia mohla fungovať ako jedna stránka bez potreby smerovania.

7.2.3 Spracovanie QR kódu

Podstatou aplikácie je využiť QR kódy pridelené zariadeniam, ktoré ich jednoznačne identifikujú. Identifikátor je obmedzený len na veľké písmená a čísla. V prípade referenčného zariadenia 8.2, QR kód obsahuje len DevEUI v hexadecimálnej podobe so 16 znakmi. Podľa LoRaWAN Aliancie [35], QR kód na LoRa zariadení musí spĺňať tri podmienky: QR kód musí byť prečítateľný čítačom, minimálna plocha QR kódu môže byť 7x7mm a musí byť možné efektívne pridávať QR kód pri masovej výrobe.

Hlavnou úlohou aplikácie je zrýchliť a zjednodušiť narábanie so zariadeniami, čo sa rieši prostredníctvom kamery. Kamerou v aplikácii užívateľ zosníma QR kód zariadenia 7.3a, pričom po úspešnom načítaní je užívateľ informovaný prostredníctvom notifikácie. V aplikácii je skenovanie implementované využitím komponentu `zxing-scanner` z knižnice `ngx-scanner`.

```
<zxing-scanner
  #scanner
  [(device)]="currentDevice"
  (scanSuccess)="onCodeResult($event)"
  (permissionResponse)="onHasPermission($event)"
></zxing-scanner>
```

Jednotlivé prvky komponentu sú:

- `[(device)]` – predstavuje zariadenie (kameru), ktorým sa bude snímať obraz,
- `(scanSuccess)` – znamená zavolanie funkcie `onCodeResult($event)` vtedy, keď sa úspešne nasníma QR kód,
- `(permissionResponse)` – spracuje informáciu o tom, či užívateľ pridelil povolenie aplikácii používať kameru.

Komponent skenera je prepojený s kódom funkcie `onCodeResult($event)` na spracovanie QR kódu.

```

onCodeResult(resultString: string): void {
    this.qrResult = resultString;
    this.gotScan.emit(resultString);
    this.onSuccess(this.qrResult)
}

```

Podstatou funkcie `onCodeResult` je v rámci komponenty uložiť reťazec spracovaného QR kódu. Stlačením tlačidla *Register Device* sa zavolá funkcia `registerNewDevice()`, ktorá zahŕňa nasledovné funkcie:

- `RHF1S001_validate_code(code)` – overí kód podľa regulárneho výrazu a vráti booleanovú hodnotu,
- `RHF1S001_registerIdentityServer(code:string)` – vytvorí a vráti objekt s dátami zariadenia pre IS,
- `RHF1S001_registerJoinServer(code:string)` – vytvorí a vráti objekt s dátami zariadenia pre JS,
- `RHF1S001_registerNetworkServer(code:string)` – vytvorí a vráti objekt s dátami zariadenia pre NS,
- `RHF1S001_registerApplicationServer(code:string)` – vytvorí a vráti objekt s dátami zariadenia pre AS.

V tejto chvíli je QR kód uložený v *scanner* komponente ako štyri rôzne objekty. `registerNewDevice()` tiež volá funkcie na prenos dáta z aplikácie na server. Jednotlivé volania *tts.reg** sú funkcie zo služby `the-things-stack.service.ts`. Tieto funkcie zahrňujú premenu objektov na reťazce formátu JSON, ktoré sa budú odosielať. Najprv sa dotazujeme na Identity Server a až tak na zvyšné servery. `subscribe()` dovoľuje asynchrónne prijať odpoveď z TTS cloudu pre prípad zachytenia chyby.

```

this.tts.regIS(this.ISData, this.currentApplication).subscribe((v) => {
    console.info(v)
    this.tts.regJS(this.JSData, devID, this.currentApplication).subscribe(
        (v) => console.info(v))
    this.tts.regNS(this.NSData, devID, this.currentApplication).subscribe(
        (v) => console.info(v))
    this.tts.regAS(this.ASData, devID, this.currentApplication).subscribe(
        (v) => console.info(v))
    this.onRegister(this.qrResult + " registered successfully")
})

```

Užívateľ je informovaný po úspešnej registrácii zariadenia 7.3b. Na stránke TTS cloudu po obnovení možno vidieť, že toto zariadenie bolo skutočne pridané 7.3c.

7.3 Užívateľské rozhranie

Užívateľské rozhranie aplikácie sa skladá z komponentov. Komponenty sú základným prvkom frameworku Angular, ktoré sú samostatne vykreslené. Vďaka tomu pri zmene na užívateľskom rozhraní sa prekreslí len daný komponent a nie celá stránka. Pozostávajú z programovej triedy (*class*), HTML šablóny a súboru CSS (Cascading Style Sheets). Výsledkom je

responzivnejšia a rýchlejšia aplikácia. Táto aplikácia používa niekoľko komponentov, najvýznamnejšie sú: *dashboard*, ktorý predstavuje úvodnú stranu so zariadeniami, *manual*, ktorý reprezentuje rozhranie manuálneho vloženia kódu užívateľom a *scanner*, ktorý predstavuje stránku s možnosťou snímania z kamery. Na vrchu aplikácie sa nachádza *header* komponent, s ktorým sa používateľ naviguje v aplikácii. Jeho úlohou je zobrazovať a skrývať dané komponenty, ktoré sú *Login*, *Devices*, *Manual* a *Scanner*. *Login* komponenta predstavuje len zastúpenie miesta, nakoľko Logimic používa svoj vlastný prihlasovací modul, s ktorým som nemal prístup použiť.

7.3.1 Dashboard komponent

Úvodná stránka aplikácie 7.2, ktorá obsahuje zoznam zariadení, ktoré môže užívateľ naskenovať do siete. Výberom zariadenia sa nastavuje TTS cloud aplikácia, ktorou zariadenie bude riadené. Jednotlivé zariadenia sú komponenty, ktoré obsahujú obrázok, názov a popis zariadenia. Následne si užívateľ vyberie manuálny alebo skenovací proces pridávania. Užívateľ si v tomto procese môže zariadenie kedykoľvek zmeniť.



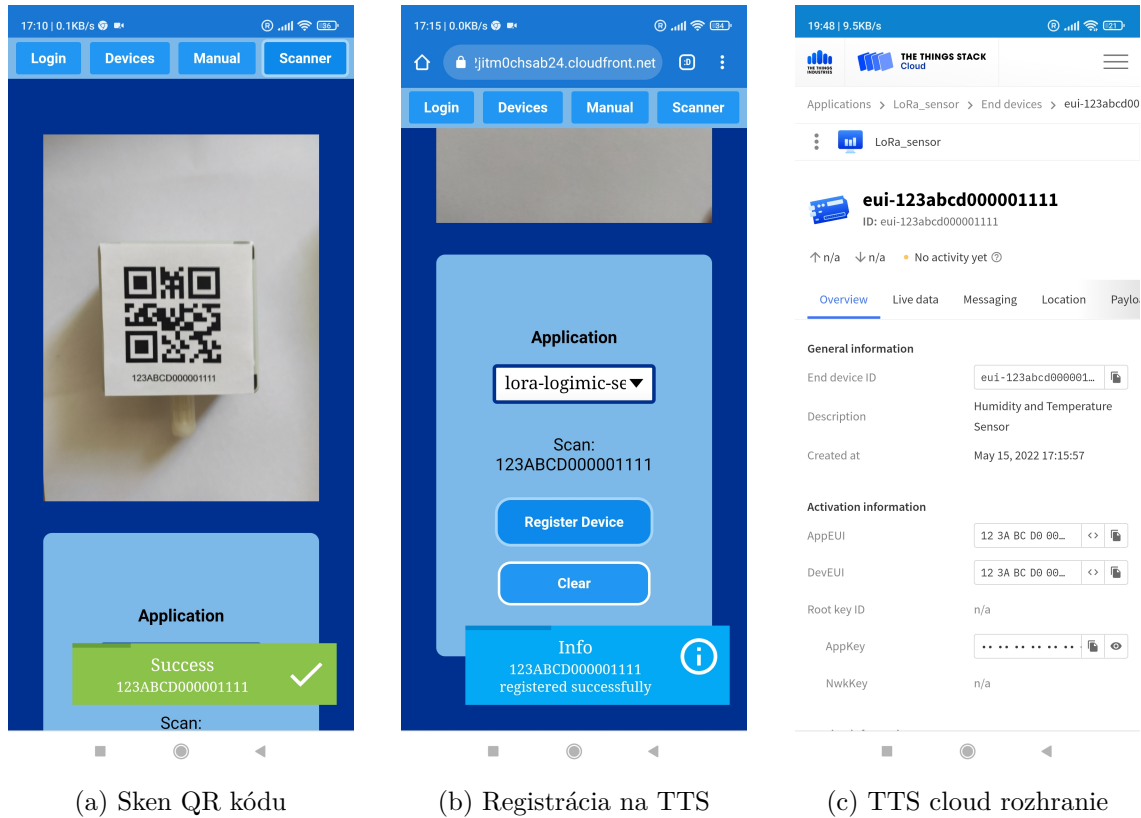
Obrázek 7.2: Výber LoRa zariadení

7.3.2 Scanner komponent

Komponent *scanner* je jadrom aplikácie. Vytvára užívateľské rozhranie pre snímanie a odosielanie dát. Z užívateľského hľadiska tu používateľ strávi najviac času. Skladá sa z *scanner* komponentu, *application-selection* komponentu a dvoch tlačidiel.

Kamerou svojho zariadenia v aplikácii užívateľ zosníma QR kód 7.3a. Po úspešnom načítaní je užívateľ informovaný prostredníctvom notifikácie. Keď užívateľ stlačí *Register*

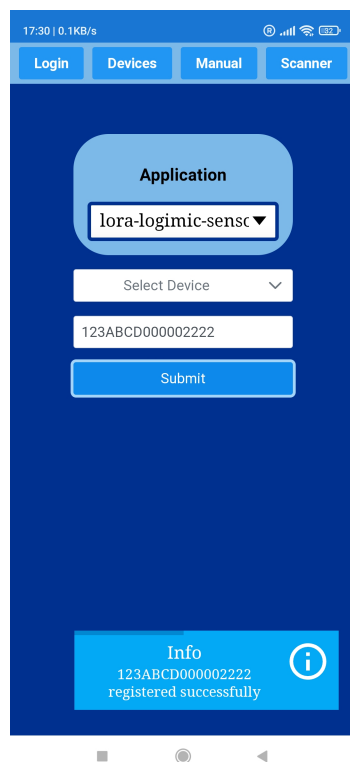
Device tlačidlo, aplikácia daný kód overí, vytvorí z neho registračné súbory vo formáte JSON a odošle ich TTS cloud. Užívateľ je informovaný po úspešnej registrácii zariadenia 7.3b. Na stránke TTS cloudu po obnovení možno vidieť, že toto zariadenie bolo skutočne pridané 7.3c.



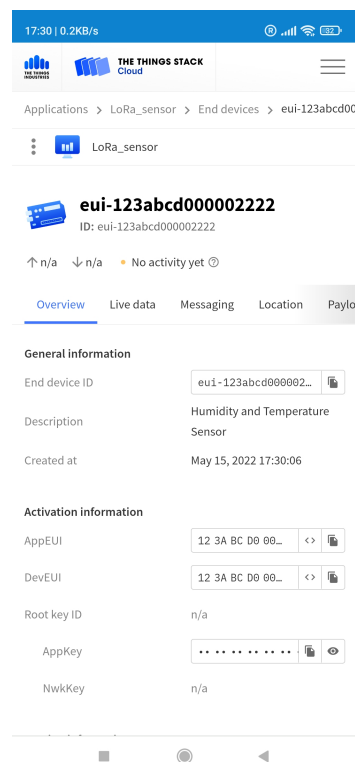
Obrázek 7.3: Proces skenovania zariadenia

7.3.3 Manuálny proces

V situácií, kedy nie je možné naskenovať QR kód, užívateľ má k dispozícii manuálne zadať informácie o zariadení a odoslať ich na TTS cloud 7.4a. Funkcionálne pracuje rovnako ako proces skenovania. Užívateľ môže zmeniť LoRa aplikáciu pre pridávané zariadenie a v prípade napísania kódu v nesprávnom formáte je užívateľ upozornený. Po registrovaní zariadenia môže prejsť späť na proces skenovania alebo naďalej skenovať manuálne. Zariadenia sú taktiež pridané automaticky po registrácii na TTS cloud 7.4b



(a) Zadanie kódu užívateľom



(b) TTS cloud rozhranie

Obrázek 7.4: Manuálne pridanie zariadenia

Kapitola 8

Testovanie

Aplikácia je nasadená na AWS cloud front sieti¹. Počas vývoja aplikácia bola testovaná vedúcim Michalom Valným, zastupujúcim firmu Logimic. Prostredníctvom jeho spätnej väzby sa v aplikácii odhalovali a riešili nedostatky.

8.1 Aplikačné testovanie

Počas implementácie boli hlavnými zameraniami kontrola a funkčnosť skeneru, splnenie podmienok PWA a korektná komunikácia a pridávanie zariadení na TTS cloud.

8.1.1 Skener

Testovanie skeneru prebiehalo najprv prostredníctvom web kamery z webového prehliadača. Očakávaným výsledkom bolo rozpoznanie žiadosti aplikácie o použitie web kamery a následné vyvolanie notifikácie z webového prehliadača o zapnutí kamery. Po splnení tohto bodu sa kontrolovala schopnosť rozpoznať a zachytávať QR kódy na zariadeniach. Z testovania vysvitlo niekoľko faktorov ovplyvňujúcich skener:

1. QR kód musí byť dostatočne osvetlený, aby ho bolo možné zosnímať.
2. QR kód nesmie byť poškodený alebo nesmie byť iná prekážka medzi kamerou a QR kódom.
3. Veľkosť QR kódu ovplyvňuje schopnosť aplikácie zosnímať kód, napríklad kód s veľkosťou strany 3cm bol príliš malý na úspešné zachytenie, teda väčšie kódy mali výhodu.
4. Zaostrenie a kvalita kamery ovplyvnila schopnosť zosnímať kód. Pri veľmi nízkej kvalite sa skener stal nespoľahlivým.
5. Vzdialenosť kamery od QR kódu výrazne ovplyvnila proces skenovania. Ideálna vzdialenosť QR kódu od kamery je 15cm a viac.
6. Doba, za ktorú skener dokáže v ideálnych podmienkach zosnímať QR kód je pod 1 sekundu, pričom ale ak sú podmienky kompromitované, môže trvať dlhšie.

Z testovania skeneru vyplynulo, že za určitých podmienok je schopný efektívne snímať QR kódy zariadení.

¹<https://d2jitm0chsab24.cloudfront.net/>

8.1.2 PWA

Pri správnej implementácii konceptu PWA bolo očakávané, že po nasadení produkčnej verzie aplikácie bude aplikácia schopná nainštalovať service worker do prehliadača a pridať možnosť inštalovať aplikáciu na užívateľské zariadenie. Otestované zariadenia mali systém Android a macOS, pričom na obidvoch sa podarilo aplikáciu úspešne nainštalovať z prehliadača. Následne sa zopakovalo testovanie skenera tentoraz pre telefóny. Vo výnimočných situáciách aplikácia neotvorila kameru zariadenia, ale pri reštartovaní aplikácie sa kamera znovu zapla a bola schopná snímania. Testy snímania dopadli úspešne, ale bol kladený väčší dôraz na vzdialenosť kamery od QR kódu. Ako PWA sa aplikácia správala rýchlo a responzívne, a zachovala si aj funkcionality offline, samozrejme okrem prípadu posielania dát na TTS cloud.

8.1.3 TTS cloud

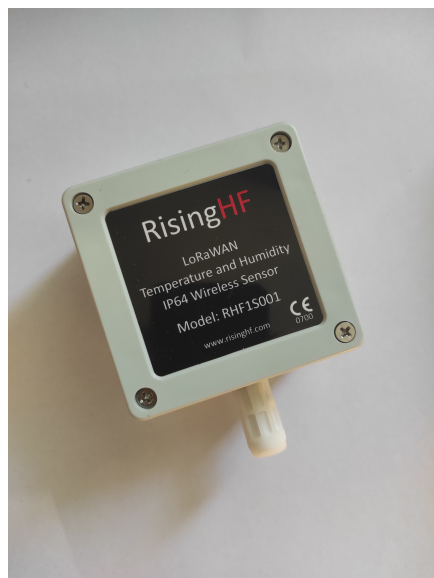
Pri testovaní komunikácie s TTS cloudom sa zistilo, že zo štyroch prístupných TTS *clusters* povoľuje APIs komunikáciu len jeden, *eu1*, nachádzajúci sa v Dubline, Írsku. Testovanie muselo prebehnúť opatrne, vzhľadom na to, že pri zaslaní chybných dát alebo použití nesprávnej metódy, cloud mohol splniť požiadavku len čiastočne alebo chybné, čo by vyvolalo komplikácie so správou zariadení. Dôkladnou prípravou a naštudovaním sa predišlo týmto problémom mohla sa otestovať komunikácia. Hlavnými zameraniami testovania boli:

1. Skontrolovať skutočné pridanie zariadenia na TTS cloud oskenovaním z aplikácie.
2. Kontrola pridania viacerých zariadení počas jednej relácie aplikácie.
3. Schopnosť spracovať chybovú hlášku pri naskenovaní jedného zariadenia viackrát.
4. Schopnosť spracovať chybu pri naskenovaní QR kódu v nesprávnom formáte.

Aplikácia splnila vyššie spomenuté testy a úspešne spracovávala dáta a komunikovala s TTS cloudom.

8.2 Referenčné zariadenie

Zariadenie, ktoré slúžilo pri vývoji aplikácie je RisingHF model RHF1S001, bezdrôtový LoRaWAN senzor na meranie teploty a vlhkosti v prostredí. Je navrhnutý na nízku spotrebu energie s životnosťou do 5 rokov bez potreby údržby. Kryt so stupňom ochrany IP64 zabráni úplnému preniknutiu prachu a proti ostrekovaním vody zo všetkých uhlov, čo ho robí vhodným na vonkajšie použitie. Senzor patrí do Triedy A. Logimic poskytol tento senzor 8.1 ako referenčné zariadenie pri práci na aplikácií. Zadná strana senzoru 8.1b obsahuje informácie a QR kód o zariadení, ktorý aplikácia využíva na pridanie senzoru do LoRaWAN siete. DevEUI je kvôli svojej unikátnosti zakryté.



(a) Predná strana senzoru



(b) Zadná strana senzoru

Obrázek 8.1: Senzor RisingHF model RHF1S0011

8.3 Návrhy na zlepšenie

Na základe spätnej väzby užívateľské rozhranie má veľa priestoru k zlepšeniu pre uspokojivú užívateľskú skúsenosť. Prihlasovaciu formu do aplikácie nebolo možné otestovať, nakoľko Logimic používa svoj vlastný modul na autentifikáciu používateľov, ku ktorému som nemal prístup. Z testovania vyplynul návrh na úpravu skenera, aby dokázal snímať i menšie QR kódy. Ďalší návrh týkajúci sa PWA je rozšírenie aplikácie na podporu lokálneho perzistentného dátového úložiska, do ktorého by sa ukladali zosnímané zariadenia a následne po pripojení na internet by sa mohli registrovať všetky naraz. Komunikácia by taktiež mohla byť rozšírená o tvorbu záznamov o zariadeniach a posielanie týchto dát do databázy o evidencii alebo podobnej databázy.

8.4 Zhrnutie

K stanoveným požiadavkám práce bolo vytvoriť aplikáciu, ktorá bude rýchlo a efektívne spracovávať LoRa zariadenia. Využitím snímania kamery sa načítajú QR kódy do aplikácie a pošlú na TTS cloud, čím sa originálny spôsob pridávania koncových zariadení stal nepotrebným. Aplikácia nie je závislá na hardvéri a využitím PWA technológie sa stala ľahko prenosná, udržateľná a lacná aplikácia s jednotným kódom. Aplikácia dokáže komunikovať s TTS cloudom, čím sa nepreruší tok dát od zariadenia na LoRaWAN server a užívateľ nie je nútený robiť prácu navyše.

Kapitola 9

Záver

Cieľom tejto bakalárskej práce bolo navrhnuť a implementovať aplikáciu na prácu s LoRa zariadeniami, ktorá by nebola závislá na hardvéri a dokázala zapojiť alebo aspoň uľahčiť pridávanie týchto zariadení do LoRaWAN siete.

Práca zahrnula naštudovanie a oboznámenie sa s relatívne novou LoRa technológiou, porozumenie problematike a využití tejto technológie v praxi. Preskúmaním rôznych cloudových LoRa platforiem a získaním referenčného zariadenia a brány som dokázal vytvoriť si malú LoRaWAN sieť a pozorovať teóriu a architektúru siete v praxi. Oboznámením sa s platformou The Things Stack cloud som porozumel spôsobu nasadzovania zariadení do siete a problematike, ktorú téma práce chcela riešiť.

Implementácia sa nezaobišla bez rôznych výziev, ktoré mi prehĺbili poznatky v tejto tematike a naučili pracovať s novými technológiami. Stretával som sa s pozitívnymi a konštruktívnymi interakciami pri diskutovaní riešenia a testovaní aplikácie, čo kladne ovplyvnilo vývoj aplikácie.

Táto práca má značný potenciál v zlepšení efektivity práce s LoRa technológiou a preto plánujem pokračovať vo vývoji a rozšíriť funkcionality aplikácie, naďalej v spolupráci s firmou Logimic. V aplikácii možno ďalej implementovať a rozvinúť offline funkcionality, rozšíriť komunikáciu o vytváranie záznamov pre tretiu stranu a mnohé iné návrhy. Verím, že táto práca priniesla hodnotu do skúmanej problematiky, a bude prakticky použiteľná klientmi firmy Logimic. Práca bola taktiež publikovaná študentskej konferencii Excel@FIT [34] pod názvom *Tool for Easy Device Configuration on LoRaWAN*.

Literatura

- [1] ADOLPH, C. *Progressive Web Apps: The future of the Mobile Web* [ebook]. Awwwards. Dostupné z: https://www.thinkwithgoogle.com/_qs/documents/6758/pwas-the-future-of-the-mobile-web.pdf.
- [2] BROWN, T. *What are IoT devices* [online]. [cit. 2022-05-05]. Dostupné z: <https://itchronicles.com/iot/what-are-iot-devices/>.
- [3] COPPEN, R. *MQTT Version 5.0* [online]. [cit. 2022-05-05]. Dostupné z: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>.
- [4] GREENGARD, S. *The Internet of Things*. MIT Press, 2015. ISBN 978-0-262-32894-4.
- [5] HEYONG WANG, F.-K. W. Enterprise cloud service architectures. *Information Technology and Management*. 2012, sv. 13. DOI: 10.1007/s10799-012-0139-4.
- [6] HUNKELER, U., TRUONG, H. L. a STANFORD CLARK, A. MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks. In: *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*. 2008, s. 791–798. DOI: 10.1109/COMSWA.2008.4554519.
- [7] JOVANOVIĆ, B. *Internet of Things statistics for 2022* [online]. [cit. 2022-05-05]. Dostupné z: <https://dataprot.net/statistics/iot-statistics/>.
- [8] KARWATKA PIOTR, W. A. *Top 30 Progressive Web Apps: Benchmarking Study* [ebook]. Divante. Dostupné z: <https://www.divante.com/reports/top-30-progressive-web-apps>.
- [9] KIM, D.-H., LEE, E.-K. a KIM, J. Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed. *Sustainability*. Březen 2019, sv. 11, s. 1917. DOI: 10.3390/su11071917. Dostupné z: https://www.researchgate.net/figure/The-LoRaWAN-MAC-protocol-stack-is-implemented-on-top-of-LoRa-modulation-PHY-The_fig1_332151302.
- [10] KJENDAL, D. *RP002-1.0.3 LoRaWAN® Regional 41 Parameters* [online]. [cit. 2022-05-06]. Dostupné z: <https://loro-alliance.org/wp-content/uploads/2021/05/RP002-1.0.3-FINAL-1.pdf>.
- [11] LAVRIC, A. a POPA, V. Internet of Things and LoRa™ low-power wide-area networks challenges. In: *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. 2017, s. 1–4. DOI: 10.1109/ECAI.2017.8166405.

- [12] LONGO, E., REDONDI, A. E., CESANA, M. a MANZONI, P. BORDER: a Benchmarking Framework for Distributed MQTT Brokers. *IEEE Internet of Things Journal*. 2022. DOI: 10.1109/JIOT.2022.3155872.
- [13] *An In-depth look at LoRaWAN® Class A Devices* [online]. [cit. 2022-05-04]. Dostupné z: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lorawan-class-a-devices/>.
- [14] *LoRa® and LoRaWAN®* [online]. [cit. 2021-12-13]. Dostupné z: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan>.
- [15] MIRAZ, M. H., ALI, M., EXCELL, P. S. a PICKING, R. A review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT). In: *2015 Internet Technologies and Applications (ITA)*. 2015, s. 219–224. DOI: 10.1109/ITechA.2015.7317398.
- [16] OLAYINKA, B. *Installing the CLI* [online]. [cit. 2022-05-04]. Dostupné z: <https://www.thethingsindustries.com/docs/getting-started/cli/installing-cli/>.
- [17] OLAYINKA, B. *MAC Commands* [online]. [cit. 2022-05-04]. Dostupné z: <https://www.thethingsindustries.com/docs/getting-started/migrating/major-changes/#mac-commands>.
- [18] OLAYINKA, B. *RX1 Delay* [online]. [cit. 2022-05-04]. Dostupné z: <https://www.thethingsindustries.com/docs/getting-started/migrating/major-changes/#rx1-delay>.
- [19] PARIDA, D., BEHERA, A., NAIK, J. K., PATTANAIK, S. a NANDA, R. S. Real-time Environment Monitoring System using ESP8266 and ThingSpeak on Internet of Things Platform. In: *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. 2019. DOI: 10.1109/ICCS45141.2019.9065451.
- [20] PETE LEPAGE, F. B. *Add a web app manifest* [online]. Dostupné z: <https://web.dev/add-manifest/>.
- [21] *Introduction to Progressive Web App Architectures* [online]. Dostupné z: <https://developers.google.com/web/ilt/pwa/introduction-to-progressive-web-app-architectures>.
- [22] *Introduction to Service Worker* [online]. [cit. 2022-05-03]. Dostupné z: <https://developers.google.com/web/ilt/pwa/introduction-to-service-worker>.
- [23] *Progressive web app structure* [online]. [cit. 2022-05-03]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/App_structure.
- [24] RAY, P. P. An Introduction to Dew Computing: Definition, Concept and Implications. *IEEE Access*. 2018, sv. 6. DOI: 10.1109/ACCESS.2017.2775042.
- [25] SAM RICHARD, P. L. *What are Progressive Web Apps?* [online]. [cit. 2022-05-03]. Dostupné z: <https://web.dev/what-are-pwas/>.

- [26] SARASWAT, M. a TRIPATHI, R. Cloud Computing: Analysis of Top 5 CSPs in SaaS, PaaS and IaaS Platforms. In: *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*. 2020, s. 300–305. DOI: 10.1109/SMART50582.2020.9337157.
- [27] SELIMOVIĆ, N. *Adding Devices* [online]. [cit. 2022-05-04]. Dostupné z: <https://www.thethingsindustries.com/docs/devices/adding-devices/>.
- [28] SELIMOVIĆ, N. *API* [online]. [cit. 2022-05-03]. Dostupné z: <https://www.thethingsindustries.com/docs/reference/api>.
- [29] SELIMOVIĆ, N. *End Device APIs* [online]. [cit. 2022-05-03]. Dostupné z: https://www.thethingsindustries.com/docs/reference/api/end_device/.
- [30] SLATS, L. *A Brief History of LoRa®: Three Inventors Share Their Personal Story at The Things Conference* [online]. [cit. 2022-05-05]. Dostupné z: <https://blog.semtech.com/a-brief-history-of-lora-three-inventors-share-their-personal-story-at-the-things-conference>.
- [31] *Number of publicly known Internet of Things (IoT) platforms worldwide from 2015 to 2019* [online]. [cit. 2022-05-04]. Dostupné z: <https://www.statista.com/statistics/1101483/global-number-iot-platform/>.
- [32] STOKKING, J. *What is new in LoRaWAN® 1.1?* [online]. [cit. 2022-05-04]. Dostupné z: <https://www.thethingsnetwork.org/article/what-is-new-in-lorawan-11-live-webinar-by-johan-stokking>.
- [33] THE THINGS NETWORK GLOBAL TEAM. *What's new in LoRaWAN® 1.0.4?* [online]. [cit. 2022-05-04]. Dostupné z: <https://www.thethingsnetwork.org/article/whats-new-in-lorawan-104-1>.
- [34] TOMÁŠ, M. Tool for Easy Device Configuration on LoRaWAN. *Sborník Excel@FIT 2022*. 2022. Dostupné z: <https://excel.fit.vutbr.cz/submissions/2022/021/21.pdf>.
- [35] YEGIN A., S. O. *LoRaWAN® Device Identification QR Codes for Automated Onboarding Technical Recommendation*. 2020. Dostupné z: https://lora-alliance.org/wp-content/uploads/2020/11/TR005_LoRaWAN_Device_Identification_QR_Codes.pdf.