

ŽILINSKÁ UNIVERZITA V ŽILINE

**FAKULTA ELEKTROTECHNIKY
A INFORMAČNÝCH TECHNOLOGIÍ**

Katedra multimédií a informačno-komunikačných technológií

28260120212064

**NÁVRH KOMUNIKAČNÉHO SYSTÉMU
PRE POSKYTNUTIE DÁT O NABÍJANÍ
ELEKTROMOBILU DO CLOUDU A
SPROSTREDKOVANIE KONCOVÉMU
POUŽÍVATEĽOVI**



ŽILINSKÁ UNIVERZITA V ŽILINE

**FAKULTA ELEKTROTECHNIKY A INFORMAČNÝCH
TECHNOLÓGIÍ**

Katedra multimédií a informačno-komunikačných technológií

**NÁVRH KOMUNIKAČNÉHO SYSTÉMU PRE POSKYTNUTIE DÁT
O NABÍJANÍ ELEKTROMOBILU DO CLOUDU A
SPROSTREDKOVANIE KONCOVÉMU POUŽÍVATEĽOVI**

DIPLOMOVÁ PRÁCA

Študijný odbor:	informatika
Študijný program:	multimediálne inžinierstvo
Vedúci diplomovej práce:	prof. Ing. Róbert Hudec, PhD.
Pracovisko vedúceho práce:	Katedra multimédií a informačno-komunikačných technológií, Žilinská univerzita v Žiline
Konzultant:	Ing. Ondrej Kasaj
Pracovisko konzultanta:	GlobalLogic Slovakia s.r.o.

Žilina, 2021

Bc. Martin Timko



Akademický rok 2020/2021

ZADANIE DIPLOMOVEJ PRÁCE

Meno, priezvisko: **Bc. Martin Timko**

Študijný odbor: **informatika**
Študijný program: **multimediálne inžinierstvo**

Téma diplomovej práce: **Návrh komunikačného systému pre poskytnutie dát o nabíjaní elektromobilu do cloudu a sprostredkovanie koncovému používateľovi**


Pokyny na vypracovanie diplomovej práce:

1. Teoreticky spracujte problematiku bezdrôtových komunikačných technológií vhodných pre e-Car IoT riešenia. Urobte prieskum štandardov v oblasti komunikačných protokolov určených pre automobily.
2. Navrhňte, vytvorte a otestujte komunikačnú architektúru pre pripojenie telematickej jednotky s cloudom. Implementačná a testovacia fáza riešenia predpokladá použitie HW simulátora telematickej jednotky.
3. Vytvorte jednoduché demo v podobe SW riešenia/notifikácie o stave nabíjania pre koncového používateľa.

Vedúci diplomovej práce: Hudec Róbert, prof. Ing. PhD., Katedra multimédií a informačno-komunikačných technológií, FEIT, ŽU v Žiline

Dátum odovzdania diplomovej práce: 03. 05. 2021

ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA ELEKTROTECHNIKY
A INFORMAČNÝCH TECHNOLOGIÍ
KATEDRA MULTIMÉDIÍ A INFORMAČNO-
KOMUNIKAČNÝCH TECHNOLOGIÍ
Univerzitná 8215/1, 010 26 Žilina


prof. Ing. Róbert Hudec, PhD.
vedúci katedry

V Žiline 30. októbra 2020

ANOTAČNÝ ZÁZNAM

Žilinská univerzita v Žiline

Fakulta elektrotechniky a informačných technológií

Katedra multimédií a informačno-komunikačných technológií

Typ práce:	Diplomová práca
Meno a priezvisko:	Bc. Martin Timko
Akademický rok:	2020/2021
Názov práce:	Návrh komunikačného systému pre poskytnutie dát o nabíjaní elektromobilu do cloudu a sprostredkovanie koncovému používateľovi
Počet strán:	105
Počet obrázkov:	42
Počet tabuliek:	3
Počet grafov:	0
Počet príloh:	4
Počet použitej lit.:	47

Anotácia v slovenskom jazyku:

Cieľom práce je analyzovať dostupné technológie umožňujúce pripojenie moderných dopravných prostriedkov do cloudu, a vytvorenie kompletného systému pre poskytovanie informácií o nabíjaní elektromobilu jeho majiteľovi. Využitá je IoT prenosová technológia, backend služby sa nachádza v cloude a sprostredkovanie údajov používateľovi prebieha pomocou mobilnej aplikácie.

Annotation in foreign language:

The aim of the thesis is to analyze available technologies enabling the connection of modern personal vehicles to the cloud, and to create a complete system for providing information about vehicle charging to its owner. IoT communication technology is used, backend is hosted in the cloud and presentation of information is carried out by using a mobile application.

Kľúčové slová: IoT, Sigfox, inteligentné vozidlá, elektromobilita, cloud

Vedúci diplomovej práce: prof. Ing. Róbert Hudec, PhD.

Konzultant DP: Ing. Ondrej Kasaj

Dátum odovzdania DP: 06. mája 2021

POĎAKOVANIE

Chcem sa poďakovať Ondrejovi Kasajovi za stálu ochotu pomôcť, záujem a trpezlivosť pri tvorení práce, a vedúcemu práce prof. Ing. Róbertovi Hudecovi, PhD. za jej odborné zhodnotenie a pripomienky.

ABSTRAKT

Abstrakt v slovenskom jazyku

TIMKO, Martin: Návrh komunikačného systému pre poskytnutie dát o nabíjaní elektromobilu do cloudu a sprostredkovanie koncovému používateľovi. [Diplomová práca]. – Žilinská univerzita v Žiline. Fakulta elektrotechniky a informačných technológií; Katedra multimédií a informačno-komunikačných technológií. – prof. Ing. Róbert Hudec, PhD. – Stupeň odbornej kvalifikácie: inžinier. – Žilina: FEIT UNIZA, 2021. 105s

Práca sa venuje vytvoreniu prehľadu a porovnaniu moderných komunikačných technológií pre pripojenie IoT zariadení do internetu a možnostiam prevádzky backendu IoT aplikácií v cloudovom prostredí. Praktickou časťou je vytvorenie systému, ktorým môžu inteligentné osobné dopravné prostriedky, predovšetkým elektromobily komunikovať s ich majiteľom, v čase keď sa pri nich bezprostredne nenachádza. Súčasťou je tiež simulátor telematickej jednotky vozidla a vytvorenie mobilnej aplikácie, ktorá slúži ako klient pre prístup k službe. Navrhnutý systém využíva možnosti moderných technológií, ako napríklad komunikačné siete pre Internet of Things a umiestnenie výpočtových kapacít služby do cloudu, čím je dosiahnutá geografická nezávislosť od lokality, kde je služba využívaná.

Kľúčové slová: IoT, Sigfox, inteligentné vozidlá, elektromobilita, cloud

Abstract in foreign language

The thesis deals with the creation of an overview and comparison of modern communication technologies dedicated to connecting IoT devices to the Internet and the possibilities of operating backend of IoT applications in a cloud environment. The practical part is about creating a system by which smart personal vehicles, especially electric cars can communicate with their owner at a time when he is not immediately present near them. The part of it is also a simulator of the vehicle's telematics unit and the creation of a mobile application that serves as a client to access this service. The proposed system uses the possibilities of modern technologies, such as communication networks for the Internet of Things and placing the computing capacity of the service into the cloud, thus achieving an geographical independence from the locality, where the service is being used.

Keywords: IoT, Sigfox, Smart vehicles, Electromobility, Cloud

OBSAH

Anotačný záznam	i
Pod'akovanie	ii
Abstrakt	iii
Abstrakt v slovenskom jazyku	iii
Abstract in foreign language	iv
Obsah	v
Zoznam obrázkov a tabuliek	vii
Zoznam skratiek	ix
Slovník termínov	xi
1 ÚVOD	1
1.1 Internet of Things	3
1.2 Cloud computing.....	5
1.3 Smart Vehicles	6
2 TELEMATICKÁ JEDNOTKA	10
2.1 Hardvér simulátora	10
2.2 Softvér simulátora.....	14
2.3 Qt GUI	17
3 KOMUNIKAČNÉ TECHNOLOGIE	19
3.1 LPWAN technológie.....	19
3.1.1 Celulárne (NB-IoT, LTE-M)	20
3.1.2 Ultra-Narrowband (Sigfox).....	24
3.1.3 Spread Spectrum (LoRa)	25
3.1.4 Telegram Splitting	32
3.2 Širokopásmové mobilné technológie.....	33
3.2.1 2G a 4G.....	34
3.2.2 5G.....	35
3.3 Lokálne pripojenie WLAN	37
3.3.1 802.11b/g/n	37
3.3.2 802.11ah a 802.11af.....	38
3.3.3 802.11p a 802.11bd.....	39
3.4 Sigfox.....	39
3.4.1 Rádiové konfigurácie	41
3.4.2 Sigfox Monarch	42

3.4.3	Dátový formát a komunikácia.....	43
3.4.4	Zabezpečenie	45
3.4.5	Dosah a pokrytie	46
3.4.6	Sigfox Atlas	48
3.4.7	Sigfox Cloud	49
3.5	Porovnanie vhodnosti pre EV a navrhnuté riešenie.....	53
4	CLOUD BACKEND	58
4.1	Sigfox Cloud	63
4.2	Amazon Web Services	63
4.2.1	Výpočtové prostredie	64
4.2.2	Úložiskové a databázové možnosti.....	67
4.2.3	API.....	70
4.3	Google Firebase	72
4.3.1	Authentication.....	72
4.3.2	Cloud Firestore a Cloud Storage.....	73
4.3.3	Cloud Messaging	73
5	KLIENTSKÁ APLIKÁCIA.....	76
5.1	Android	76
5.2	Návrh aplikácie	78
5.3	Komunikácia	82
5.4	Ukladanie dát	82
6	ZÁVER	84
	Zoznam použitej literatúry	86
	Čestné vyhlásenie	89
	Prílohová časť	90
	Zoznam príloh.....	I

ZOZNAM OBRÁZKOV A TABULIEK

Obrázky:

Obr. 1.1: Ilustrácia prepojenia množstva zariadení v rámci IoT	4
Obr. 1.2: Tlačidlo aktivácie eCall a obrazovka infotainmentu vozidla	7
Obr. 1.3: Koncept praktickej časti práce.....	9
Obr. 2.1: Mikropočítač Raspberry Pi 4B samostatne a v použitej krabicike s displejom	11
Obr. 2.2: Výstupy zbernice GPIO na Raspberry Pi	11
Obr. 2.3: Použitý komunikačný modul Sigfox	12
Obr. 2.4: Ukážka príkazov komunikačného protokolu Sigfox modulu.....	13
Obr. 2.5: Príklad priebehu nabíjania batérie elektromobilu.....	14
Obr. 2.6: Rozloženie prenášaných údajov do paketov.....	16
Obr. 2.7: Grafické rozhranie simulátora	18
Obr. 3.1: Režimy prevádzky NB-IoT	22
Obr. 3.2: Porovnanie vlastností LTE-M a NB-IoT.....	24
Obr. 3.3: Rozdelenie technológie LoRa vo vrstvách.....	27
Obr. 3.4: Architektúra systému LoRa.....	28
Obr. 3.5: Outdoorový LoRa gateway od spoločnosti Friendcom.....	29
Obr. 3.6: Vývojový LoRa komunikačný modul	29
Obr. 3.7: Časový vývoj generácií mobilných sietí.....	34
Obr. 3.8: Architektúra systému Sigfox	40
Obr. 3.9: Sigfox modem a vývojársky USB modul.....	40
Obr. 3.10: Mapa regiónov technológie Sigfox	41
Obr. 3.11: Pokrytie Sigfox siete celosvetovo	47
Obr. 3.12: Pokrytie Sigfox siete na Slovensku	47
Obr. 3.13: Jednotka mikrozakladňovej stanice Sigfoxu a repeater signálu	48
Obr. 3.14: Prehľad prijatých správ zo zariadenia v Sigfox Backende.....	50
Obr. 3.15: Gramatika dekodovacej syntaxe.....	51
Obr. 3.16: Ukážka callback metódy v Sigfox Backende	52
Obr. 3.17: Logá štyroch najväčších technológií z oblasti LPWAN.....	53
Obr. 4.1: Porovnanie distribučných modelov cloudu a On-premises riešenia.....	59
Obr. 4.2: Rozdelenie trhu cloudových prevádzkovateľov (2020) [37].....	61
Obr. 4.3: Dekódovacia gramatika regulárnej správy z vozidla.....	63
Obr. 4.4: Rôzne prístupy poskytnutia výpočtových kapacít v cloud-computingu.....	66

Obr. 4.5: Ilustrácia základných typov NoSQL databáz	68
Obr. 4.6: Ukážka záznamu stavu vozidla v DynamoDB databáze	69
Obr. 4.7: Konfigurácia a diagram API metódy pre príjem aktualizácie	71
Obr. 4.8: Architektúra služby Firebase Cloud Messaging.....	75
Obr. 5.1: Architektúra operačného systému Android	77
Obr. 5.2: Hlavná obrazovka a selektor vozidla.....	79
Obr. 5.3: Obrazovka histórie a selektor dátumu a času	80
Obr. 5.4: Obrazovky nastavení a zobrazenia mapy	81
Obr. 5.5: Výrez dát z tabuľky stavov vozidiel	83
Obr. 5.6: Údaje ukladané do SharedPreferences	83
Obr. 6.1: Architektúra navrhnutého systému a použité technológie.....	84

Tabuľky:

Tab. 3.1: Parametre a špecifiká technológie Sigfox v regiónoch	41
Tab. 3.2: Dĺžka vysielania pri rôznych dĺžkach správy	43
Tab. 3.3: Formát uplink Sigfox správy	44

ZOZNAM SKRATIEK

Skratka	Anglický význam	Slovenský význam
3GPP	The 3rd Generation Partnership Project	-
ACID	Atomicity, Consistency, Isolation, Durability	Atomicita, Konzistencia, Izolácia, Trvalosť
API	Application Programming Interface	Aplikačné programové rozhranie
AWS	Amazon Web Services	-
BTS	Base Transceiver Station	Základňová vysielaco-prijímacia stanica
CAN	Controller Area Network	-
CRC	Cyclic Redundancy Check	Kontrola cyklickým kódom
CSS	Chirp Spread Spectrum	-
CSMA	Carrier Sense Multiple Access	Metóda viacnásobného prístupu
DBPSK	Differential Binary Phase-Shift Keying	Rozdielové binárne kľúčovanie fázovým posunom
DSL	Digital Subscriber Line	Digitálna účastnícka prípojka
ECU	Electronic Control Unit	Elektronická riadiaca jednotka
EIRP	Equivalent Isotropically Radiated Power	Ekvivalentný izotropne vyžiarený výkon
ETSI	European Telecommunications Standards Institute	Európsky inštitút pre telekomunikačné normy
EUI	Extended Unique Identifier	Rozšírený unikátny identifikátor
EV	Electric Vehicle	Elektromobil
FaaS	Function-as-a-Service	Funkcia ako služba
FDMA	Frequency Division Multiple Access	Viacnásobný prístup frekvenčným delením
FEC	Forward Error Correction	Dopredná chybová korekcia
GCP	Google Cloud Platform	-
GFSK	Gaussian Frequency-Shift Keying	Gaussovo kľúčovanie frekvenčným posunom
GND	Ground	Elektrická zem
GPIO	General Purpose Input/Output	-
GPS	Global Positioning System	Globálny lokalizačný systém
GSM	Global System for Mobile Communications	Globálny systém mobilných komunikácií
GUI	Graphical User Interface	Grafické používateľské rozhranie
HDD	Hard Disk Drive	Pevný disk
HTTP	Hypertext Transfer Protocol	Hypertextový prenosový protokol

HW	Hardware	Hardvér
ID	Identifier	Identifikátor
IDE	Integrated Development Environment	Integrované vývojárske rozhranie
I2C	Inter-Integrated Circuit	-
I/O	Input/Output	Vstup/výstup
ITU	International Telecommunication Union	Medzinárodná telekomunikačná únia
IEEE	Institute of Electrical and Electronics Engineers	Inštitút elektrotechnického a elektronického inžinierstva
IoT	Internet of Things	Internet vecí
IIoT	Industrial IoT	Priemyselné IoT
ISM	Industrial, Scientific and Medical	Nelicencované frekvenčné pásmo
IT	Information Technology	Informačné technológie
JSON	JavaScript Object Notation	-
LAN	Local Area Network	Lokálna sieť
LCD	Liquid Crystal Display	Displej s kvapalnými kryštálmi
LPWAN	Low-Power Wide-Area Network	Nízkoenergetická veľko-rozlohová sieť
LTE	Long Term Evolution	-
M2M	Machine-To-Machine	Komunikácia medzi strojmi
MAC	Media Access Control	Fyzická adresa sieťového zariadenia (všeobecne)
MAC	Message Authentication Code	Autentifikačný kód správ (technológia Sigfox)
MIMO	Multiple-Input Multiple-Output	Viacnásobný vstup, viacnásobný výstup
MVVM	Model-View-ViewModel	-
OS	Operating System	Operačný systém
OTA	Over-The-Air	operácia na diaľku, bezdrôtovo
PAC	Porting Authorization Code	Autorizačný kód prenosu
POI	Point of Interest	Bod záujmu
QAM	Quadrature Amplitude Modulation	Kvadratúrna amplitúdová modulácia
QPSK	Quadrature Phase Shift Keying	Kvadratúrne kľúčovanie fázovým posunom
RAM	Random Access Memory	Pamäť náhodného prístupu
REST	Representational State Transfer	-
RF	Radio Frequency	Rádiová frekvencia
RSSI	Received Signal Strength Indicator	Indikátor sily prijatého signálu
SDK	Software Development Kit	Sada nástrojov pre vývoj softvéru

SIM	Subscriber Identity Module	Modul identifikácie účastníka
SLA	Service-Level Agreement	Dohoda o úrovni poskytovaných služieb
SMS	Short Message System	System krátkych textových správ
SSD	Solid State Drive	Polovodičový disk
SW	Software	Softvér
TCU	Telematic Control Unit	Telemetrická riadiaca jednotka
TFT	Thin Filter Transistor	-
TPM	Trusted Platform Module	Modul zabezpečujúci platformu
UART	Universal Asynchronous Receiver and Transmitter	Jednotka sériového komunikačného rozhrania
UI	User Interface	Používateľské rozhranie
UID	Unique Identifier	Unikátny identifikátor
UX	User Experience	Používateľský zážitok
UNB	Ultra Narrow Band	System využívajúci extrémne úzke frekvenčné spektrum
URI	Uniform Resource Locator	Jednotný identifikátor zdroja
URL	Uniform Resource Locator	Jednotný ukazovateľ zdroja
USB	Universal Serial Bus	Univerzálna sériová zbernica
VLAN	Virtual LAN	Virtuálna lokálna sieť
VoLTE	Voice-over-LTE	Prenos hlasu sieťou LTE
V2V	Vehicle-To-Vehicle	Komunikácia typu vozidlo-vozidlo
V2X	Vehicle-To-Everything	Komunikácia typu vozidlo-čokoľvek
VPN	Virtual Private Network	Virtuálna privátna sieť
WAN	Wide Area Network	Sieť na rozľahlom území
WLAN	Wireless LAN	Bezdrôtová lokálna sieť
WPA	Wi-Fi Protected Access	Zabezpečený prístup k Wi-Fi

*Niektoré skratky nie sú preložené, pretože nemajú v slovenčine ekvivalentné pomenovanie alebo ide o názov.

SLOVNÍK TERMÍNOV

Termín	Význam termínu
Automotive	Oblasť priemyslu zameraná na vývoj a výrobu automobilov
Access-point	Zariadenie fungujúce ako prístupový bod do bezdrôtovej siete
Backend	Infraštruktúra, ktorá umožňuje poskytovanie nejakej služby

Beacon	Statické oznamovacie alebo synchronizačné vysielanie
Cachovanie	Ukladanie údajov do dočasnej pamäte
Callback	Mechanizmus na volanie nejakej entity pri nastaní nejakej udalosti
Cloud	Serverová infraštruktúra v internete poskytujúca služby klientom
Cloud-computing	Výpočtové služby prevádzkované v cloudovom prostredí
Cluster	Zoskupenie serverov v logickom alebo geografickom zväzku
Dashboard	Ovládací/prehľadový panel
Downlink	Komunikácia v zostupnom smere, zo servera na klienta
Duty cycle	Pracovný cyklus
Endianita	Poradie ukladania bajtov dátového typu v pamäti počítača
Embedded (systém)	Vstavaný systém, ktorý je súčasťou komplexnejšieho systému
Endpoint	Koncový, resp. začiatkový bod rozhrania služby pre komunikáciu s inou službou
Firmvér	Riadiaci softvér zariadenia
Framework	Balík softvérových knižníc určených na podporu vývoja nového softvéru
Frontend	Časť softvérového riešenia, s ktorou užívateľ prichádza do styku
Handshake	Proces dohodnutia parametrov komunikácie pred prenosom dát
Hardvér	Všetko čo sa nedá ukradnúť cez internet
Hosting	Prenájom kapacity pre prevádzkovanie služby
Infotainment	Palubný systém vozidla umožňujúci užívateľovi interakciu s vozidlom a poskytujúci multimediálne funkcie
Multiplexovanie	Súčasný prenos viacerých signálov jedným médiom
On-premises	Prostredie vlastnené prevádzkovateľom služby
Open-source	Licencia voľnej dostupnosti
Paket	Balík informácií, dát, prenášaných sieťou alebo zbernicou
Payload	Obsah prenášanej správy, bez hlavičky a pätičky
Pin	Konektor pre jednoduché pripojenie jedného vodiča
Prerušenie	Funkcia, ktorej spustenie je vyvolané nejakou udalosťou
Roaming	Poskytovanie telekomunikačnej služby mimo domovskej siete poskytovateľa
Runtime	Behové prostredie softvéru
Smartfón	Mobilný telefón s pokročilým operačným systémom umožňujúcim inštalovať externé služby a programy
Softvér	Vybavenie počítača do ktorého sa nedá kopnúť
Uplink	Komunikácia vo vzostupnom smere, z klienta na server
Watchdog	Mechanizmus dohliadajúci na funkčnosť zariadenia alebo služby
Wi-Fi	Rodina štandardov bezdrôtových komunikačných technológií

1 ÚVOD

V súčasnej dobe sledujeme v technologickom svete 2 veľké fenomény – prvým, v oblasti informačných technológií, je mohutné začínanie používania cloudových riešení, teda presun výpočtových aj úložiskových kapacít najrôznejších služieb na serverové farmy vlastnené a prevádzkované spoločnosťami tretích strán; a druhý, prevažne v automobilovom priemysle, je postupná zmena dominantného spôsobu pohonu dopravných prostriedkov zo spaľovacích motorov na elektrické motory, ktoré neprodukujú žiadne emisie, poskytujú často lepšie výkonové parametre a v neposlednom rade sú konštrukčne oveľa jednoduchšie. Keďže svet moderných technológií je veľmi pestrý a jednotlivé odvetvia sa navzájom prelínajú, kombinujú a podporujú za cieľom priniesť používateľom čo najlepší možný zážitok, nemali by sme zabúdať ani na kombináciu spomenutých dvoch fenoménov a možnosti, ktoré sa tým otvárajú. Všetky významné technologické spoločnosti sa dnes snažia pridávať do svojich produktov nejakú pridanú hodnotu, funkcie, ktoré činia zariadenia a služby “chytrými“ – umožňujú im integrovať sa s inými službami či zdrojmi údajov, komunikovať spolu, vyhodnocovať rôzne situácie a reagovať na ne.

Do tohto konceptu perfektne zapadá aj myšlienka ponúknuť osobným dopravným prostriedkom funkcionalitu jednoducho a efektívne komunikovať s ich majiteľom a poskytnúť mu rôzne informácie o sebe, v dobe keď sú mu užitočné. Medzi tieto dopravné prostriedky môžeme zaradiť nielen osobné autá, ale tiež motocykle, bicykle, či napríklad v poslednej dobe veľmi na popularite rastúce elektrické kolobežky. Túto myšlienku ešte viac podporuje spomínaný nárast elektromobility, kedy všetky tieto zariadenia prechádzajú na elektrický pohon so zdrojom energie z batérií, ktoré treba dobíjať. Používateľ často necháva svoj dopravný prostriedok nabíjať aj mimo domova, či už na rôznych verejných nabíjačkách na strategických miestach, v práci, alebo napríklad na klasických čerpacích staniaciach vybavených aj nabíjacími stanicami. Je pritom predpoklad, že majiteľ nie je po celú dobu nabíjania v bezprostrednej blízkosti vozidla – môže využiť čas napríklad na nakupovanie alebo ísť do budovy stanice na kávu. V tejto chvíli nastáva situácia kedy potrebuje vedieť ako nabíjanie pokračuje – koľko času zostáva alebo aký dojazd bude mať k dispozícii ak by vyrazil teraz. Medzi ďalšie možnosti môžeme zaradiť prezeranie si informácií o doterajšej spotrebe, o teplote vo

vozidle alebo jeho okolí a následne vzdialené ovládanie klimatizácie alebo vykurovania, upozornenie na nebezpečne vysokú teplotu a súčasnú prítomnosť domáceho zvierat'a, či možno zistiť polohu prostriedku v prípade krádeže. Kľúčová požadovaná vlastnosť je jednoduchosť tohto procesu pre používateľa, čo znamená najmä odstránenie/potlačenie problémov s konektivitou zariadení – najmä vozidla. Nemôžeme očakávať, že majiteľ bude po pripojení na nabíjačku riešiť ešte pripájanie vozidla na nejakú blízku Wi-Fi sieť. Taktiež sa žiada zabezpečiť kontinuálnu funkčnosť systému aj pri cestovaní do zahraničia. To značne sťažuje aj použitie bežných mobilných sietí, pretože by bolo potrebné riešiť medzinárodný roaming a veci s tým spojené. Riešenie ponúkajú moderné IoT komunikačné siete, z ktorých vybrané vedia fungovať nezávisle od operátora ktorý ich prevádzkuje, sú lacné, spoľahlivé a poskytujú dostatočné pokrytie väčšiny zaľudnených miest v krajinách rozvinutého sveta.

Cieľom práce je porovnať dostupné technológie a na základe týchto požiadaviek vytvoriť systém, ktorý bude možné integrovať do rôznych typov dopravných prostriedkov (s dôrazom na elektromobily), a ktorý umožní jednoduchý a spoľahlivý prenos želaných informácií do cloudu, kde môžu byť vyhodnotené a uschované, a následne do smartfónu majiteľa a ich prezentovanie prostredníctvom mobilnej aplikácie.

Práca je rozdelená na niekoľko celkov podľa logickej postupnosti, ako vytvorený systém funguje. V úvode je zoznámenie sa s fenoménmi IoT a cloudových technológií, a s požiadavkami na moderné inteligentné vozidlá. Pokračuje druhou kapitolou, ktorá sa venuje telematickej jednotke v dopravnom prostriedku, ktorá slúži na zber údajov. Pre účely tejto práce je telematická jednotka nahradená simulátorom postavenom na mikropočítači. Tretia kapitola sa venuje komunikačným technológiám vhodným pre prenos údajov z IoT zariadení do cloudového prostredia, ich porovnaníu, a hlbšie rozoberá zvolenú technológiu Sigfox. V štvrtej kapitole je rozobraná problematika cloudov, počnúc backendom technológie Sigfox, pokračujúc cez verejný cloud AWS na ktorom beží logika služby navrhnutého systému, po platformu Google Firebase, ktorej služby sme použili v rámci mobilnej aplikácie. Posledná piata kapitola je o tejto mobilnej aplikácii, ktorú má používateľ – majiteľ dopravných prostriedkov nainštalovanú vo svojom smartfóne a ktorá mu prezentuje obdržané údaje.

1.1 Internet of Things

Termín Internet of Things (IoT), alebo Internet vecí predstavuje filozofiu pripojenia obrovského množstva *vecí* do *internetu*, za účelom zdieľania najrôznejších údajov a/alebo vykonávania akcií na základe okolností.

Internet v tomto prípade nemusí znamenať internet ako ho poznáme (sieť založená na adresovaní IP). Môže ísť aj o siete fungujúce na inej báze, ktoré sú do internetu pripojené prostredníctvom nejakej brány. Pojem internet je teda myslený v pôvodnom význame slova – “Interconnected Networks“ – vzájomne prepojené siete.

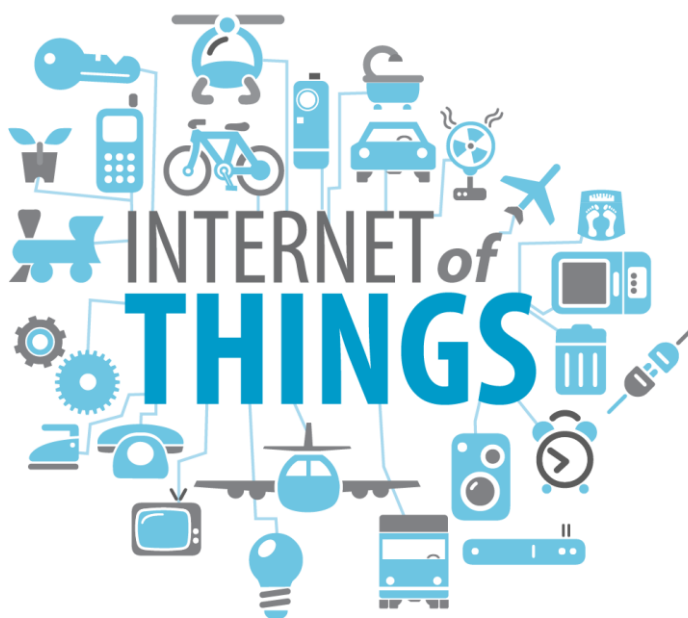
Veci predstavujú najrôznejšie zariadenia, ktoré existujú vo svete okolo nás. Pôvodne pritom nemusí ísť ani o elektronické zariadenia, no pridaním senzoru na sledovanie nejakej vlastnosti alebo nejakého spôsobu regulácie sa stávajú kandidátmi zapojiť sa do tejto filozofie. Na nízkej úrovni môžeme tieto veci rozdeliť na 2 skupiny – senzory, ktoré do siete dáta poskytujú, a aktuátory (tzv. akčné členy), ktoré na základe vyhodnotených dát vykonávajú nejakú akciu.

Skôr ako pokúšať sa o presnú definíciu IoT je vhodnejšie povedať, že ide o fenomén, ktorý pokrýva množstvo rôznych oblastí – od osobných vecí, cez domácnosti, mestá, zdravotníctvo, poľnohospodárstvo, dopravu a logistiku, po priemysel a výrobu. Cieľmi sú najčastejšie optimalizácia úloh, zlepšovanie pohodlia, úspora energií, či zvyšovanie bezpečnosti.

Bežný človek sa s pojmom IoT stretol najpravdepodobnejšie v oblasti inteligentných domácností (smart home). Veci ako žiarovky, termostaty, žalúzie, zámky, audio systém, atď. môžu byť ovládané automaticky na základe dát z meteorologických staníc, senzorov osvetlenia, pohybu, videa z kamier alebo celkom externých údajov z rôznych služieb, alebo manuálne, cez lokálnu sieť alebo na diaľku cez internet. Dá sa však pokračovať ďalej, do sveta IoT patria aj chytré televízory, hlasoví asistenti, robotické vysávače, alebo aj inteligentné chladničky, práčky či rýchlovarné kanvice. Mestá začínajú zavádzať IoT technológie pre ovládanie pouličného osvetlenia, monitorovania kvality ovzdušia, naplnenosti odpadových kontajnerov, obsadenosti parkovacích miest, kontrolu hydrantov, automatický diaľkový odpočet meračov v bytovkách, či k efektívnemu riadeniu dopravy. Táto oblasť sa nazýva smart city. V medicínskej oblasti ide napríklad o monitorovanie pacientov, v poľnohospodárske monitorovanie plodín. Do sveta IoT

môžeme tiež zaradiť osobné zariadenia ako fitness náramky, bezdrôtové slúchadlá s ovládaním, a nakoniec samozrejme samotné smartfóny, ktoré vlastní takmer každý človek.

Skratka IIoT znamená IoT v oblasti priemyslu, písmeno I navyše je zo slova Industrial. Aplikácie v tejto oblasti pokrývajú monitorovanie a riadenie strojov v továrňach – ovládanie robotov, prepojenie PLC automatov a výrobných liniek, tiež oblasť energetiky a stavebníctva. Keďže mnoho strojov bolo do nejakej formy siete pripojené už skôr, je nemožné povedať, kde IoT “začína“. Dá sa ale povedať, že to čo do priemyslu prináša je efektívnejšie sledovanie a riadenie výrobných procesov a ekonomické úspory. Pokrok v priemysle, ktorý moderné technológie vrátane riešení IIoT umožňujú sa súhrne označuje aj ako štvrtá priemyselná revolúcia – Industry 4.0.



Obr. 1.1: Ilustrácia prepojenia množstva zariadení v rámci IoT

Nie je možné určiť hranice čo IoT je a čo už/ešte nie. Základnou myšlienkou ale je, že aplikácia zhromažďuje údaje zo senzorov a na základe týchto údajov môže iniciovať určité akcie. Okrem toho je možné údaje použiť na analýzu, aby sa získali rôzne užitočné informácie a prehľady. S tým súvisí prepojenie IoT s ďalšími oblasťami – AI (umelou inteligenciou), ML (strojovým učením) a Big data, pre analýzu veľkého množstva nazbieraných údajov.

Jedným z hlavných rozhodnutí, ktoré sa musia pri vytváraní IoT aplikácií prijať, je aká prenosová technológia pre pripojenie do internetu bude pre dané použitie najvhodnejšia. Väčšinou ide o bezdrôtové pripojenie. V súčasnosti na nachádzame v dobe, kedy vzniká pre potreby IoT množstvo nových komunikačných technológií, formujú sa, niektoré navzájom súťažia, iné si vôbec nekonkurujú, a mnohé aj zanikajú. Oblasti IoT sú tak široké, že je nemožné vytvoriť jednu technológiu, ktorá by spĺňala požiadavky všetkých aplikácií. Niekde je potrebná komunikácia na malú vzdialenosť v rozsahu miestnosti, inde sa vyžaduje vonkajšie pokrytie na území celých krajín. Niekedy úplne postačuje rýchlosť pripojenia na úrovni desiatok bajtov za sekundu, inokedy je potrebný prenos videa vo vysokom rozlíšení. Rôzne krajiny sveta majú tiež rozdielne navrhnuté plány využívania frekvenčného spektra a iné legislatívne regulácie. Oboznámeniu sa s rôznymi bezdrôtovými komunikačnými technológiami pre svet IoT, hlavne vhodnými pre oblasť automotive, sa venuje kapitola 3.

1.2 Cloud computing

Hoci pojmy cloud a cloud-computing sú pre väčšinu ľudí taktiež nejasné, v tomto prípade je definícia pomerne jednoduchá. Po tom, čo sa údaje z IoT zariadení dostanú do internetu, putujú na nejakú serverovú infraštruktúru, kde sú ukladané a vyhodnocované. Túto serverovú infraštruktúru môže vlastniť priamo prevádzkovateľ služby, alebo tretia strana, od ktorej sú služby v nejakej podobe prenajímané. Podľa toho hovoríme o súkromných alebo verejných cloudoch. Spoločnosti ktoré prevádzkujú verejné cloudy a ich služby prenajímajú sa nazývajú prevádzkovatelia cloudových služieb. Tento prenájom môže mať množstvo rôznych podôb – môže byť prenajímaná celá fyzická infraštruktúra, virtuálna platforma, konkrétna aplikácia, či iný logický celok. Pre označovanie predmetu služby je zaužívaný akronymom XaaS (X-as-a-Service) a súhrne sa označujú ako cloud-computing. Pojem cloud je potom chápaný všeobecne ako virtuálny priestor v ktorom sú poskytované služby cloud-computingu, alebo je použitý pre označenie riešenia od konkrétneho poskytovateľa. Oblasť cloud-computingu a jeho využitie v našom riešení podrobnejšie rozoberá kapitola 4.

1.3 Smart Vehicles

Pre túto prácu je najpodstatnejšia oblasť automotive, o ktorej sa v súvislosti s IoT zatiaľ až tak veľa nehovorí, no prirodzene do nej patrí. Moderné automobily sú vybavené elektronickými systémami, ktoré prinášajú množstvo užitočných funkcií. Ak pridáme vozidlu možnosť konektivity do internetu alebo k iným sieťam, otvára sa mnoho nových možností pre ďalšie funkcionality a služby [1]:

Údaje pre infotainment, navigáciu, autopilota

Často veľká obrazovka infotainmentu je veľmi pohodlná pre prezeranie mapových podkladov so satelitnými snímkami, vyhľadávania POI bodov v okolí alebo aj na prehliadanie webstránok. Pre elektromobily je zvlášť prínosné vyhľadávanie nabíjacích staníc v okolí a zistenie ich obsadenosti. Súčasťou infotainmentu môže byť tiež hlasový asistent. Navigačný systém môže oveľa efektívnejšie plánovať trasu a upozorňovať na riziká na ceste, pokiaľ má online informácie. Stále zlepšujúce sa systémy autopilotov potrebujú zdieľať údaje so servermi za účelom neustáleho tréningu neurónových sietí. Všetky tieto služby potrebujú pre svoje fungovanie plnohodnotnú konektivitu k internetu.

OTA aktualizácie

Ide o spôsob aktualizácií softvérových častí automobilu na diaľku (Over-The-Air). Môže ísť o aktualizácie firmvéru riadiacich jednotiek, softvéru infotainmentu, mapových podkladov navigačného systému, ale aj sprístupňovanie celkom nových funkcií, ktoré výrobca vyvinul dodatočne alebo ktoré si majiteľ aktivoval ako doplatkovú výbavu.

eCall

eCall je systém automatického núdzového volania na tiesňovú linku v prípade vážnej dopravnej nehody. V prípade, že riadiaca jednotka systému vyhodnotí kolíziu, napríklad vystrelením airbagov alebo prudkou zmenou hybnosti vozidla, automaticky pomocou nezávislej GSM jednotky kontaktuje tiesňovú linku a odošle údaje o presnej polohe vozidla, doterajšom smere jazdy, identifikáciu a prípadne ďalšie podrobnosti (napr. mieru poškodenia vozidla). Systém môže byť tiež prepojený s audiosystémom vo vozidle pre umožnenie hlasového spojenia s operátorom tiesňovej linky. V Európskej únii je systém eCall štandardizovaný, poskytovaný bezplatne a od roku 2018 je povinnou výbavou každého nového vozidla [2].



Obr. 1.2: Tlačidlo aktivácie eCall a obrazovka infotainmentu vozidla

Vzdialené ovládanie

Pokiaľ má vozidlo dostupnú stálu konektivitu aj vo vypnutom stave, je možné to využiť na poskytnutie vzdialeného ovládania. Hoci ide zatiaľ o výsadu prémiových modelov, je takto možné pomocou aplikácie v smartfóne ovládať napríklad vykurovanie/klimatizáciu a pripraviť si príjemnú teplotu vo vozidle ešte pred cestou, zmeniť nastavenia nabíjania alebo si privolať auto z parkovacieho miesta, ak podporuje dostatočnú úroveň autonómneho riadenia.

Alarmy na udalosti

Pokiaľ má používateľ vo svojom smartfóne nainštalovanú aplikáciu, prostredníctvom ktorej ho môže automobil kontaktovať v ľubovoľnom čase, je možné ho takto upozorniť na neštandardné udalosti, ktoré sa v aute alebo jeho okolí v neprítomnosti majiteľa stanú. Môže ísť napríklad o zabudnuté domáce zvieratá alebo dieťa v aute, pričom vnútorná teplota dosiahla nebezpečnú úroveň, náraz do zaparkovaného vozidla alebo podozrivú osobu pohybujúcu sa v jeho okolí.

Pokročilá ochrana pred odcudzením

Hoci odcudzenie moderných automobilov je nepravdepodobné, pri iných dopravných prostriedkoch je to stále problém. Konektivita s dopravným prostriedkom dokáže zabezpečiť možnosť sledovať jeho polohu, na diaľku ho zablokovať a spustiť alarm. Navyše niektoré IoT komunikačné technológie ponúkajú vysokú odolnosť proti rušeniu, teda je predpoklad, že aspoň obmedzené spojenie zostane dostupné aj napriek snahe zlodca aktívne rušiť signál.

In-car konektivita

Pod týmto pojmom rozumieme vzájomné prepojenie inteligentných zariadení vnútri vozidla, napríklad za účelom zdieľania multimediálneho obsahu alebo internetového pripojenia. Prenos informácií zo smartfónu do infotainment systému riešia napríklad systémy Android Auto a Apple CarPlay. Pripojenie do internetu môže byť zdieľané zo smartfónu vozidlu, aj naopak, z vozidla do smartfónov pasažierov.

Informácie o vozidle, manažment flotily

Medzi tieto informácie môže patriť čokoľvek, čo je majiteľovi užitočné. Vo firemnom použití, napríklad pri prepravných spoločnostiach je pre špeditérov nevyhnutné mať v reálnom čase prehľad o polohe všetkých vozidiel svojej flotily, dĺžkach ich jazd a prestávkach, spotrebe paliva a tak ďalej. Do kontextu správy flotily môžeme zaradiť aj služby zdieľania rôznych dopravných prostriedkov v mestách, ktoré v posledných rokoch veľmi naberajú na popularite. Pri súkromnom použití môžu „flotilu“ predstavovať dopravné prostriedky rôzneho typu patriace jednému majiteľovi, prípadne rodine. Používateľ – majiteľ si potom prostredníctvom aplikácie môže prezerať informácie o svojich vozidlách, napríklad o nabíjaní, čo je primárna téma tejto záverečnej práce.

V2V a V2X komunikácia

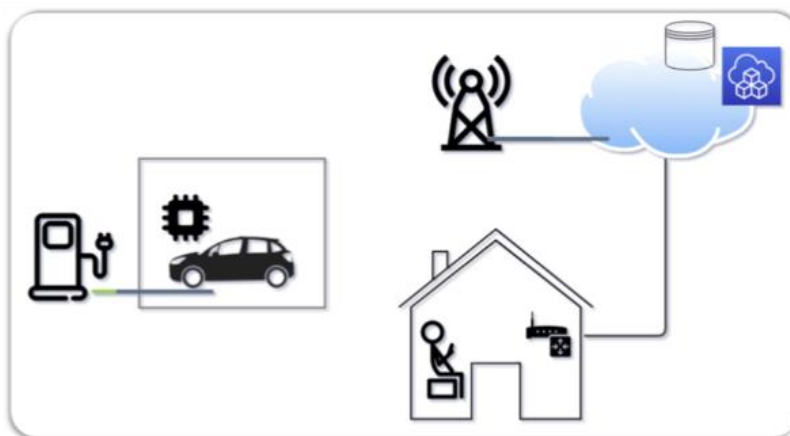
V dlhodobjšom horizonte vývoja automotive priemyslu je kľúčovým fenoménom koncept priamej komunikácie blízkych vozidiel navzájom, tzv. V2V (Vehicle-to-Vehicle) komunikácia. Autá na ceste budú medzi sebou vytvárať krátkodobé spojenia za účelom rýchlej výmeny informácií o dopravných podmienkach, prekážkach na ceste, blížiacich sa vozidlách záchranných služieb, alebo v prípade autonómneho riadenia aj o vzájomnej koordinácii – synchronizované rozbiehanie sa na križovatkách alebo v zápachach, prejazdy semaforami s kratšími rezervami pri zmenách farby, atď. Ďalšou úrovňou tejto koncepcie je V2X (Vehicle-to-Everything), kedy bude vozidlo komunikovať s akýmkoľvek objektom v jeho okolí – napríklad semaforami samotnými, systémami na monitoring dopravy, mýtnymi bránami, parkovacími a závorovými automatmi, blízkymi chodcami, a tak ďalej. Hoci ide z hľadiska nasadenia zatiaľ skôr o futuristickú myšlienku, smerovanie k tomuto trendu je jasné a už dlho existujú pre tento účel viaceré komunikačné štandardy. Všetka táto vzájomná komunikácia bude za účelom zvyšovania bezpečnosti a efektivity dopravy, úspory energií a zlepšovania pohodlia, čo

presne zodpovedá cieľom IoT samotného. Tento koncept býva niekedy pomenovaný aj pojmom IoV (Internet of Vehicles), teda Internet vozidiel.

Tieto funkcionality robia súčasné moderné a budúce vozidlá skutočne inteligentnými zariadeniami, preto sa často označujú ako Smart vehicles. Treba podotknúť, že nemusí ísť len o automobily, množstvo z týchto služieb je použiteľných aj pri iných spomínaných typoch dopravných prostriedkov.

Praktickou časťou tejto práce je vytvoriť systém pre komunikáciu práve týchto rôznych typov dopravných prostriedkov so svojim majiteľom, za účelom poskytnutia alebo vytvorenia prostredia pre poskytnutie viacerých z vyššie popísaných funkcionalít moderných inteligentných dopravných prostriedkov. Zo zadania sa práca venuje konektivite vozidla (primárne elektromobilu) pomocou nízkorýchlostnej IoT technológie (prípadne doplnenej lokálnym WLAN pripojením), ktorá nie je náhradou za plnohodnotné pripojenie do internetu prostredníctvom širokopásmového mobilného pripojenia, zato poskytuje iné výhody a špecifiká.

Základný koncept a architektúru praktickej časti práce zobrazuje obrázok č. 1.3:



Obr. 1.3: Koncept praktickej časti práce

2 TELEMATICKÁ JEDNOTKA

Telematická jednotka (TCU) je elektronická riadiaca jednotka (ECU) v automobile, resp. inom (dopravnom) prostriedku, ktorej úloha je zbierať údaje z iných riadiacich jednotiek alebo zo senzorov a zabezpečovať bezdrôtovú komunikáciu vozidla s cloudom, za účelom prenosu týchto informácií. Taktiež môže prijímať údaje alebo inštrukcie z cloudu. Táto komunikácia môže prebiehať prostredníctvom rôznych technológií, čomu sa venuje kapitola 3. Telematická jednotka je vybavená modemom pre prístup do siete konkrétnej technológie, pričom použitých ich môže byť aj viacero – napr. celulárna GSM technológia pre prenos veľkého množstva údajov doplnená nízkorychlostnou LPWAN technológiou pre základnú konektivitu s vysokou dostupnosťou. Komunikačný modul je zvyčajne pripojený na anténny systém, ktorý môže byť umiestnený na vhodnom mieste vo/na vozidle. Keďže medzi funkcie, ktorú TCU sprostredkúva môže patriť aj vzdialená aktivácia iných systémov, zostáva TCU ako jeden z mála systémov zostáva napájaná aj vo vypnutom/nenašartovanom stave automobilu [3]. V rámci tejto práce nahradíme telematickú jednotku reálneho vozidla jej simulátorom, postavenom na mikropočítači Raspberry Pi.

2.1 Hardvér simulátora

Ako hardvér, na ktorom simulátor pobeží sme sa rozhodli zvoliť mikropočítač Raspberry Pi v najnovšej verzii 4B v konfigurácii so 4GB pamäte RAM. Dôvodom tohto výberu je jednoduchosť a obľúbenosť tohto zariadenia, dostupnosť príslušenstva a dobré možnosti pripojenia periférií.

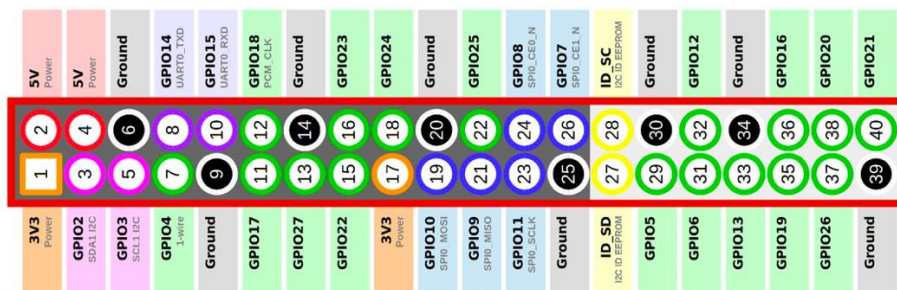
Raspberry Pi je malý počítač veľkosti približnej kreditnej karty, ktorý na jedinej doske plošného spoja obsahuje všetky súčiastky potrebné na vývoj, demonštráciu aj prevádzku riešení – procesor, pamäť RAM, Wi-Fi modul, Bluetooth modul, porty na pripojenie periférií (USB 2.0 aj 3.0, Gigabit Ethernet, 2x microHDMI a 3,5mm audio jack), a taktiež vyvedenú 40-pinovú GPIO zbernicu pre pripojenie ďalších elektrických zariadení alebo modulov s integrovanou podporou všetkých bežných komunikačných protokolov – UART, SPI a I²C. Použitý je štvorjadrový procesor ARM Cortex-A72 s výrobnou technológiou 28nm s taktovacou frekvenciou 1,5GHz a voliteľné množstvo pamäte RAM

(v našom riešení verzia s 4GB) typu LPDDR4. Operačným systémom je Linux distribúcia s názvom Raspberry Pi OS (v minulosti nazývaná Raspbian), odvodená od distribúcie Debian, alebo je možné ale použiť aj iné linuxové distribúcie. V minulosti existovala aj experimentálna podpora OS Windows vo verzii IoT Core (pre modelovú radu 3), no pre aktuálne modely nie je v čase písania práce dostupná a ani sa to neočakáva [4,5].



Obr. 2.1: Mikropočítač Raspberry Pi 4B samostatne a v použitej krabičke s displejom

Využitie výstupov zbernice GPIO môžeme vidieť na obrázku č. 2.2. Univerzálne piny majú zabudované pull-up aj pull-down rezistory, na zbernici je prítomné napájanie 5V aj 3,3V, pričom ale GPIO výstupy pracujú s 3,3V logikou.



Obr. 2.2: Výstupy zbernice GPIO na Raspberry Pi

V zariadení nášho simulátora na túto zbernicu pripájame všetok doplnkový hardvér. Ide o komunikačný modul, ktorý sa pripája na rozhranie UART (piny 8 a 10), tepelný senzor pripojený na zbernicu I²C (piny 3 a 5) a modul dotykového displeja, ktorý je pripojený pomocou zbernice SPI (piny 19, 21, 23 a 24). Okrem toho používame napájanie 3,3V (pin 1) pre komunikačný modul a tepelný senzor, 5V pre displej (pin 2), a GND (piny 6, 9, 20 a 25).

Použitý komunikačný modul na prístup do siete Sigfox (bližšie v kapitole 3) je pripojený prostredníctvom rozhrania UART (signály RXD a TXD), signálovej zeme a je napájaný napätím 3,3V. Voliteľne je možné pripojiť ešte linku pre signál WUP (Wake-Up), ktorý sa používa na zobudenie modemu z režimu spánku, no túto funkciu sme nepoužili. Na modul sa prostredníctvom mikrokoaxiálneho konektora MHF2 pripája externá anténa. V našom prípade sme použili len jednoduchú krátku anténu so ziskom 3dBi, ktorá sa vojde do vnútra krabičky, no je možné použiť anténu s ľubovoľným ziskom. Spotreba modulu počas vysielania je do 65mA, pri nečinnosti 500 μ A a pri použití režimu spánku len 2 μ A [6].



Obr. 2.3: Použitý komunikačný modul Sigfox

Parametre pre UART komunikáciu sú stanovené pevne: symbolová rýchlosť 9600baud, 8 dátových bitov, 1 stop bit, žiadna parita, bez riadenia toku. Všetka komunikácia s modulom prebieha pomocou AT príkazov, čo je zaužívaný veľmi jednoduchý formát textovej komunikácie, pri ktorom správy začínajú znakmi “AT” a sú ukončované riadiacim znakom Carriage Return (CR, 0x0D), Line Feed (LF, 0x0A), alebo oboma súčasne. Ovládací protokol pozostáva z niekoľkých príkazov na odoslanie bitu/rámca, nastavenie parametrov prenosu, systémových registrov, získanie informácií atď. Pre testovanie spojenia s modulom a jednoduché resetovanie linky v prípade chyby parsovania je možné použiť základný príkaz “AT”, na ktorý modul odpovedá “OK”.

AT command	Name	description
AT	Dummy Command	Just return 'OK' and does nothing else. Can be used to check communication.
AT\$SB=bit[,bit]	Send Bit	Send a bit status (0 or 1). Optional bit flag indicates if AX-SFEU should receive a downlink frame.
AT\$SF=frame[,bit]	Send Frame	Send payload data, 1 to 12 bytes. Optional bit flag indicates if AX-SFEU should receive a downlink frame.
AT\$SO	Manually send out of band message	Send the out-of-band message.
AT\$STR?	Get the transmit repeat	Returns the number of transmit repeats.
AT\$STR=uint	Set transmit repeat	Sets the transmit repeat.
AT\$uint?	Get Register	Query a specific configuration register's value. See chapter "Registers" for a list of registers.

Obr. 2.4: Ukážka príkazov komunikačného protokolu Sigfox modulu

Na meranie teploty je použitý multisenzor Bosch BMP280, ktorý integruje senzory na meranie teploty vzduchu a atmosférického tlaku a komunikuje prostredníctvom zbernice I²C. Senzor používa na zbernici adresu 0x76, na ktorej periodicky vysiela namerané hodnoty v 20-bitovom rámci. Rozsah merania teploty je -40~85°C pri rozlíšení 0,01°C a presnosti 1°C. Napájanie senzoru je 3,3V [7].

Modul displeja sa pripája na zbernicu SPI a je ovládaný proprietárnou knižnicou dodávanou výrobcom, ktorá sa inštaluje priamo do operačného systému Raspberry Pi OS. LCD displej je typu TFT, uhlopriečku a rozlíšenie má 3,5" a 480x320 pixelov, obnovovacia frekvencia je 50Hz. Dotyková plocha je kapacitnej technológie.

Výkon mikropočítača Raspberry Pi je výrazne vyšší ako je pre účel takejto telematickej jednotky potrebné a pri návrhu reálnej jednotky by postačoval výkon na úrovni 8bitových mikrokontrolérov a niekoľko kB operačnej pamäte¹. Použitie tejto obľúbenej platformy nám ale umožnilo jednoducho obohatiť zariadenie simulátora telematickej jednotky prídavným príslušenstvom napríklad v podobe krabičky s dotykovým kapacitným displejom, vďaka ktorej je možné simulátor veľmi jednoducho ovládať.

¹ V prípade priamej komunikácie cez IP s použitím šifrovania je nevyhnutné počítať s vyššími nárokmi a potrebou 32-bitovej architektúry.

2.2 Softvér simulátora

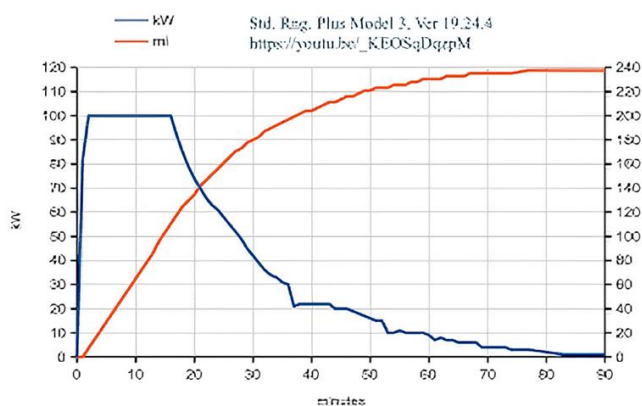
Základom simulátora je simulačný program, ktorý sme vytvorili v jazyku C++ s podporou frameworku Qt, pomocou ktorého je vytvorené GUI rozhranie pre ovládanie simulácie. Aplikácia je písaná objektovo a okrem hlavnej funkcie (main) pozostáva z viacerých komponentov, bežiacich v samostatných vláknach:

- Komponent simulátora – samotný simulátor údajov
- Komponent “broadcaster“ – zabezpečuje komprimovanie údajov, vyhodnocuje dostupnú konektivitu a odosiela údaje na potrebný komunikačný modul (HW)
- Komponent pre UI – ovládanie simulátora
- Komponent “watchdog“ – sleduje stav ostatných komponentov a v prípade ich zlyhania reštartuje službu

Výstupom simulátora je niekoľko parametrov vozidla, pri jeho rôznych stavoch – vypnuté (Off), nabíjanie (Charging), nečinné (Idle) a jazda (Driving). Medzi tieto údaje patria:

- aktuálny percentuálny stav nabitia
- aktuálne napätie batérie
- doterajšia dĺžka trvania aktuálneho stavu
- odhad zostávajúceho času (v prípade nabíjania)
- dojazd vozidla pri aktuálnom nabití a spotrebe

Základným údajom potrebným pre dopočet ďalších, je percentuálny stav nabitia, ktorého hodnota je na základe časového priebehu a zvoleného nabíjacieho prúdu vypočítavaná tak, aby simulovala exponenciálnu krivku priebehu nabíjania reálnych batérií, ktorej príklad môžeme vidieť na obrázku č. 2.5 [8]:



Obr. 2.5: Príklad priebehu nabíjania batérie elektromobilu

Simulátor poskytuje tiež niekoľko ďalších údajov, ktoré sú ale konštantné, len pre účely ukážky – vonkajšia teplota pri vozidle, poloha a spotreba elektriny, a navyše vnútornú teplotu vo vozidle, ktorá je vyčítavaná z tepelného senzora pripojeného k Raspberry Pi.

Aplikácia beží v operačnom systéme Raspberry Pi OS ako služba zavedená cez inicializačný systém systemd, čo umožní jej automatické spustenie po naboťovaní zariadenia a prípadné resetovanie v prípade chyby. Po spustení procesu hlavná funkcia iniciuje hardvér, načíta zo súboru údaje o vozidle (napr. maximálne napätie batérie), nastaví počiatočné hodnoty simulovaných premenných a spustí vlákna ostatných komponentov. Počiatočný stav vozidla je nečinnosť, simulovanie nabíjania je možné spustiť prepnutím do stavu nabíjanie. Správanie pre stav jazda sme neimplementovali.

Vo vlákne komponentu simulátora beží nekonečný cyklus, ktorý v pravidelných intervaloch (pre účely ukážky nastavené na 5 sekúnd) a v závislosti od nastaveného stavu vozidla vypočítava simulované hodnoty. Percentuálna hodnota aktuálneho stavu nabitia je vypočítavaná pomocou exponenciálnej funkcie a jej inverznej funkcie, popísanými vzťahmi:

$$x = \frac{1}{a} * \log\left(-\frac{y - 100}{100}\right)$$

$$y = 100 * (1 - e^{-ax})$$

kde a predstavuje koeficient rýchlosti nabíjania (závislosť od nabíjacieho prúdu), y je percentuálna hodnota nabitia z rozsahu 0-100, a x je “pozícia“ v priebehu nabíjania (relatívny čas), ktorú je potrebné vedieť vypočítať v prípade, že nabíjanie nezačína od nulovej hodnoty. Parameter a je určený lineárnou škálou medzi dvoma krajnými pevné definovanými koeficientami rýchlosti nabíjania (pri prúde 15A a 300A).

Keďže priebeh exponenciálnej funkcie prechádza v blízkosti konečnej hodnoty do limity a nikdy túto hodnotu nedosiahne, záver priebehu je orezaný podmienkou, kedy hodnota vyššia ako 99,5 je zaokrúhlená na 100. Ďalšie simulované údaje sú potom odvodené od percentuálnej hodnoty nabitia.

Komponent broadcaster má rovnako vlastný cyklus, v ktorom v pravidelných intervaloch (nastavené na 20 sekúnd) zabezpečuje proces odvysielania aktuálneho stavu

vozidla. Tento proces pozostáva najprv z komprimácie údajov do 2 typov paketov o dĺžkach 12 bajtov, nakoľko toto je maximálna veľkosť payloadu jednej správy v technológii Sigfox. Kompresia je dosiahnutá presne stanoveným bitovým posunom pri jednotlivých premenných s údajmi, nakoľko väčšina z nich nevyužíva celý rozsah hodnôt poskytnutý použitými dátovými typmi; respektíve pri údajoch s reálnym číselným typom sú premenné pretypované na neštandardný dátový typ half [9], ktorý oproti typu float (32 bitov) zaberá len 16 bitov na úkor zníženej presnosti, ktorá je ale pre naše potreby úplne dostačujúca. Prvý typ paketu (regulárny) sa posiela periodicky, druhý typ (rozšírený) je posielaný len pri zmene stavu vozidla a iných prípadoch (bližšie popísané v kapitole 3.5), keďže obsahuje menej premenlivé údaje. Rozdelenie údajov do paketov a ich štruktúru môžeme vidieť na obrázku č. 2.6:

Regular message format:

attribute:	length:	data type:	unit:
- state	(3 bits)	enum	-
- current charge	(7 bits)	uint8	perc
- target charge	(7 bits)	uint8	perc
- current	(10 bits)	uint16	amps
- elapsed time	(13 bits)	uint16	minutes
- remain. time	(13 bits)	uint16	minutes
- current range	(11 bits)	uint16	km
- outs. temperature	(16 bits)	half	°C
- ins. temperature	(16 bits)	half	°C

96 bits			
=12 bytes			

Extended message format:

attribute:	length:	data type:	unit:
- location lat	(26 bits)	half	coordinates
- location long	(26 bits)	half	coordinates
- des. temperature	(16 bits)	half	°C
- max. current	(10 bits)	uint16	amps
- elec. consumption	(16 bits)	half	kWh/100km
- reserve	(2 bits)	-	-

96 bits			
=12 bytes			

Obr. 2.6: Rozloženie prenášaných údajov do paketov

Po vytvorení paketu prebehne vyhodnotenie dostupnej konektivity (dostupnosť WLAN pripojenia). Ak je dostupné pripojenie cez Wi-Fi, odošle sa paket pomocou HTTP spojenia priamo na REST API backendu v cloude (bližšie rozobrané v kapitole 4.2), ak Wi-Fi dostupné nie je, odošle sa paket na sériový port (rozhranie UART), na ktorom ho

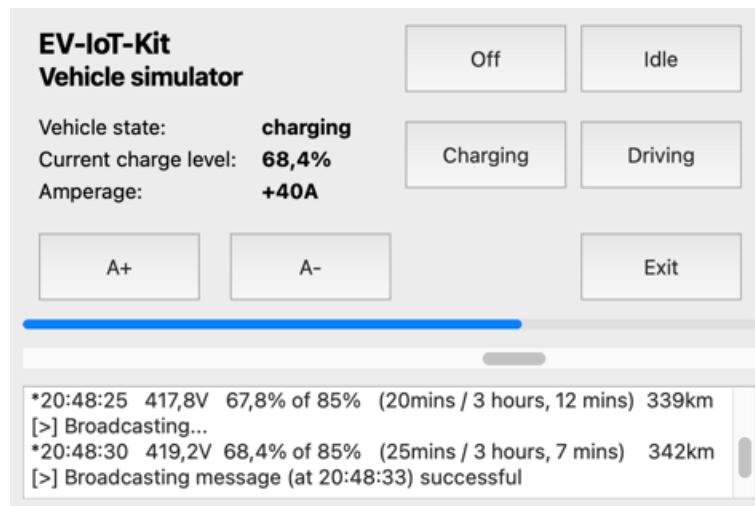
príjme a spracuje komunikačný modul. V oboch prípadoch sa vyhodnotí odpoveď siete/modulu a do výstupu simulátora sa vypíše informácia o (ne)úspešnosti odoslania.

Integrácií skutočnej riadiacej jednotky do systému vozidla sa táto práca nevenuje, je však možné predpokladať nutnosť oficiálnej integrácie výrobcom, nakoľko prístup k informáciám z palubného systému moderných automobilov, napríklad prostredníctvom zbernice CAN, je veľmi dobre zabezpečený a závislý na konkrétnom modeli vozidla alebo jeho výrobcovi.

2.3 Qt GUI

Pre vytvorenie grafického rozhrania na ovládanie simulátora sme použili obľúbený framework Qt. Ide o multiplatformný framework napísaný v jazyku C++, ktorý ponúka široké možnosti od tvorby grafického rozhrania, cez prístup k hardvéru, až po spracovanie multimediálnych dát. Qt je primárne určený pre jazyk C++, no prostredníctvom prídavných vrstiev alebo samostatných verzií je prístupný aj jazyky Python, JavaScript, Java, Ruby, C a ďalšie. Podporuje lokalizáciu aplikácií, konektivitu na SQL databázy, spracovanie XML, správu vlákien, prístup k súborom, prácu so sieťou, grafikou a multimédiami, atď. Aplikácie môžu byť skompilované pre takmer všetky existujúce platformy – Windows, Linux, macOS, Android, iOS, aj real-time operačné systémy ako napríklad WxWorks, pričom prvky GUI majú vždy natívny vzhľad operačného systému. V súčasnosti je však kvôli vplyvu konkurenčných technológií na mobilných platformách Qt používaný prevažne pre programovanie desktopových a embedded aplikácií. V automobiloch ho pre svoj infotainment používajú napríklad aj spoločnosti Tesla, BMW a Volvo [10]. Jeho veľkou výhodou je veľmi prehľadne spracovaná dokumentácia a taktiež vlastné IDE Qt Creator. Dostupný je pod open-source aj komerčnými licenciami, v závislosti od použitia [11].

Pre našu aplikáciu sme použili moduly Qt GUI a Qt Widgets, a ich základné komponenty QLabel, QPushButton, QProgressBar, QScrollBar a QPlainTextEdit, pomocou ktorých sme vytvorili jednoduché grafické rozhranie navrhnuté s ohľadom na veľkosť a rozlíšenie použitého dotykového displeja.



Obr. 2.7: Grafické rozhranie simulátora

Rozhranie ponúka pomocou tlačidiel možnosť zmeniť stav vozidla a zvýšiť/znížiť nabíjací prúd, pomocou scrollbaru je možné zmeniť aktuálnu úroveň nabitia. Tiež je možné sledovať aktuálne hodnoty simulovaných parametrov. Po kliknutí na tlačidlá sú volané callback funkcie, ktoré zmenia príslušné parametre simulácie, výstup simulátora je zobrazovaný v textovom poli v spodnej časti okna. Aplikácia sa automaticky spúšťa v celoobrazovkovom režime.

3 KOMUNIKAČNÉ TECHNOLOGIE

Ďalším potrebným článkom systému je prenosová technológia, prostredníctvom ktorej sú dáta z vozidla prenášané do cloudovej infraštruktúry. Na tento článok systému sú kladené mnohé požiadavky, vyplývajúce z rôznych funkcií ktoré má náš systém, alebo aj moderné vozidlo samo o sebe poskytovať. Niektoré z týchto predpokladov sú ťažko zlučiteľné, preto môžu byť použité aj viaceré technológie. Napríklad, infotainment vozidla môže potrebovať plnohodnotnú konektivitu k internetu pre získanie mapových podkladov a inštrukcií pre navigáciu, a to kedykoľvek počas jazdy. Na tento účel je potrebné spojenie s internetom prostredníctvom niektorej zo štandardných GSM mobilných technológií. Funkcia OTA aktualizácií softvéru vozidla zas môže potrebovať prenášať veľké objemy údajov, na čo sa javí vhodnejšie použiť pripojenie k lokálnej sieti Wi-Fi. Pre účely prenosu malého množstva telematických alebo telemetrických údajov je zas lepšie použiť LPWAN technológiu. O tom, ktorá z dostupných technológií sa použije kedy a na aký účel, rozhoduje telematická jednotka. V tejto kapitole sú analyzované rôzne možnosti konektivity IoT zariadení a v závere je posúdená ich vhodnosť pre použitie v našom riešení.

3.1 LPWAN technológie

LPWAN (Low-Power WAN) sú technológie bezdrôtových sietí pre rozsiahle územia, ktoré slúžia na prepojenie zariadení na veľké vzdialenosti nízkou prenosovou rýchlosťou. Nemusí však ísť o priame pripojenie do internetu, nemusí byť možné komunikovať na základe IP adres. Prenos dát v takýchto sieťach je pomalý, ale protiváhou je vyššia výkonová efektívnosť ako pri klasických mobilných sieťach. Boli vytvorené pre účely M2M (Machine-To-Machine) komunikácie v IoT svete, kde stroje medzi sebou nepotrebujú prenášať veľké množstvo údajov.

Výhodou nízkej rýchlosti je nízka energetická náročnosť, vďaka ktorej môžu jednoduché senzory komunikujúce pomocou týchto technológií dosiahnuť výdrž batérie až v horizontoch niekoľkých rokov. To je veľmi výhodné najmä pre jednoduché jednoúčelové zariadenia, ktoré sa vďaka tomu stávajú takmer bezúdržbové. Ak ide o zariadenie na inštaláciu v exteriéri, stačí použiť veľmi malý solárny článok a stáva sa

úplne energeticky nezávislé. Taktiež ceny použitia LPWAN sietí bývajú rádovo nižšie ako pri klasických GSM technológiách, to isté platí aj pre hardvér koncových zariadení, ktorý môže byť oveľa jednoduchší. Ďalším faktorom umožňujúcim nižšiu cenu je, že množstvo z týchto technológií pracuje vo voľnom, teda nelicencovanom frekvenčnom spektre, za ktorého využívanie nie je potrebné platiť licenčné poplatky.

Siete LPWAN bývajú založené na hviezdicovej topológii. Komunikácia môže byť jednosmerná aj obojsmerná, no predpokladá sa častejší prenos informácií z koncového zariadenia smerom k základňovej stanici (BTS). Rozloženie základňových staníc je navrhnuté tak, aby sa zóny vzájomne prekrývali. Výhodou hviezdicovej, na rozdiel od mriežkovej topológie je, že koncové zariadenia sú nezávislé na iných koncových zariadeniach. Ak sa stane, že niektoré zariadenie prestane pracovať, funkčnosť siete nie je ovplyvnená. Zariadenia môžu byť pridávané a odoberané bez toho aby ovplyvnili ostatné zariadenia v sieti. Vzhľadom k tomu, že všetky koncové zariadenia sú pripojené k základňovej stanici a tá je centrálnym prvkom topológie, celá sieť je ale závislá od jej fungovania. Ak by sa stalo, že by základňová stanica prestala fungovať, či už v prípade straty napájania alebo inej poruchy, znefunkčnilo by to konektivitu všetkých zariadení v jej dosahu, ktoré nemajú pokrytie súčasne aj z iných základňových staníc.

Z hľadiska využívaného technického princípu vieme LPWAN technológie rozdeliť na 4 nižšie podrobnejšie rozobraté kategórie, pri ktorých sú spomenutí aj ich hlavní predstavitelia [12]. Okrem spomenutých existuje viacero ďalších technológií, ktoré sa ale neuplatnili globálne, boli už nahradené inou, alebo ich vývoj/nasadzovanie ešte len prebieha.

3.1.1 Celulárne (NB-IoT, LTE-M)

Sú postavené nad technológiami klasických bunkových sietí, fungujúcimi podľa štandardov 3GPP. Sú prevádzkované v licencovanom frekvenčnom spektre vlastnenom operátorom danej siete, čo znamená veľmi vysokú spoľahlivosť, najmä vďaka faktu, že nemôže nastať rušenie žiadnymi inými systémami. Z povahy týchto štandardov vyplýva nutnosť precíznejšej časovej a frekvenčnej synchronizácie a handshake procedúry pred začiatkom vysielania. Môže byť potrebné, aby koncové zariadenie niekoľkokrát kontaktovalo základňovú stanicu, kým dostane povolenie na odvysielanie správy. Tento

prístup minimalizuje riziko kolízií pri vysielaní viacerých zariadení, ale pridáva značnú réžiu prenosu, čo zvyšuje energetickú náročnosť a znižuje jej predvídateľnosť. Ako výsledok “väčšej opatrnosti“ je možné poskytnúť vyššie prenosové rýchlosti, čo ale znova zvyšuje energetické nároky.

Technológie z tejto kategórie poskytujú väčšinou väčší operátori, ktorí ich prevádzkujú súčasne so svojimi GSM sieťami. Platí, že do týchto sietí je potrebné sa autentifikovať, čo býva zabezpečené použitím SIM kariet. Z týchto faktov vyplýva vyššia komplikovanosť celého systému, vyššia cena a možnosti roamingu obmedzené dohodami (často nadnárodných) operátorov. Ako vhodné sa tieto technológie javia pre použitie v komplexnejších aplikáciách, kde je kľúčová robustnosť systému a ktoré vedia ťažiť z vyššej dostupnej prenosovej rýchlosti. [12]

Zástupcami tejto kategórie sú napríklad technológie NB-IoT a LTE-M.

3.1.1.1 NB-IoT

NB-IoT (Narrowband IoT), technickým názvom LTE Cat NB, je jedno z mladších LPWAN riešení, ktoré vzniklo modifikáciou technológie LTE určenej pre mobilný internet (LTE Cat 4). Technológia je štandardizovaná organizáciou 3GPP (Release 13 a 14) a funguje v licencovanom pásme existujúcej mobilnej 4G infraštruktúry. Šírka komunikačného pásma sa znížila z 20MHz na 200kHz, čo zodpovedá jednému bloku v systémoch GSM. Klesla tým maximálna prenosová rýchlosť, no zlepšilo sa tým pokrytie siete o cca 20dB. Komplexnosť modemov je zhruba päťnásobná v porovnaní s modemami pre plnohodnotné LTE. Prenosová rýchlosť sa môže pohybovať od 600bit/s do cca 200kbit/s a je možná obojsmerná komunikácia, nie však súčasne. Používa sa modulácia QPSK a multiplexovanie signálov pomocou FDMA pre uplink a OFDMA (Orthogonal FDMA) pre downlink. Maximálny payload jednej správy je 512B. V sieti odpadá nutnosť brán, keďže zariadenia majú pridelené IP adresy a komunikujú priamo s internetom. Na transportnej vrstve sa používa protokol UDP. Vďaka synchronizácii v sieti sa môže v jednej bunke nachádzať až 50 000 aktívnych zariadení [13].

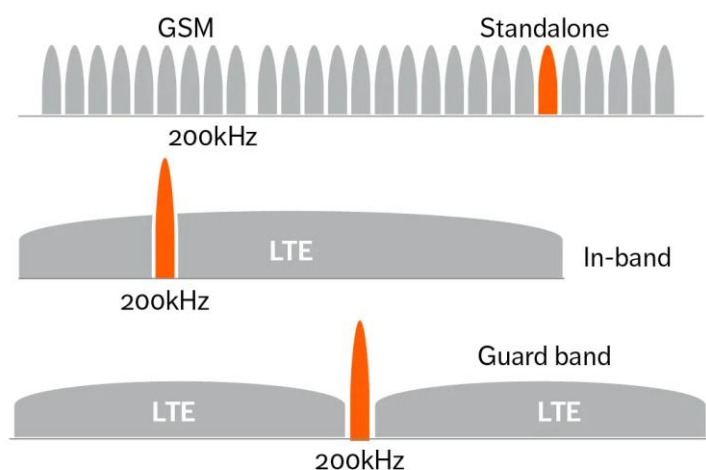
Kľúčovými výhodami sú:

- Možná dostupnosť všade tam, kde je už dostupné LTE pripojenie bez potreby budovať samostatnú sieť
- Výborné pokrytie horšie prístupných miest, ako sú podzemné priestory a šachty

- Vďaka licencovanému pásmu neobmedzený počet správ za deň

Energetická náročnosť je dobrá v oblastiach s dobrým pokrytím signálom, no je výrazne horšia pri slabšom signáli a navyše nie je dobre predvídateľná – modem sa musí čakať na povolenie vyslať. S tým súvisí aj možná dlhšia doba doručovania, ktorá sa udáva do 10 sekúnd. Nevýhodou sú tiež drahšie komunikačné moduly v porovnaní s inými technológiami a nevhodnosť pre pohyblivé aplikácie – modem potrebuje byť pripojený k jednej BTS, prepájanie je energeticky náročné a trvá relatívne dlho. Medzinárodný roaming je možný, no len vtedy, ak majú operátori uzavreté príslušné dohody. [38]

NB-IoT dokáže z hľadiska prevádzky súčasne s inými sieťami fungovať v troch módoch – samostatne (Stand-alone), kedy nie je závislý od LTE, vo vyhradenom pásme (Guard Band), kedy je mu vyhradená v pásme použítom LTE medzera, alebo priamo v spektre LTE (In-Band):



Obr. 3.1: Režimy prevádzky NB-IoT

Na Slovensku prevádzkujú technológiu NB-IoT operátori Slovak Telekom, ktorý okrem firemných riešení poskytuje túto službu aj bežným zákazníkom, a O2 Slovakia [14].

3.1.1.2 LTE-M

LTE-M (LTE Machine type) je ďalší z podstandardov technológie LTE, vyvinutý v zhruba rovnakom čase ako NB-IoT. Jeho označenie je LTE Cat M1 a bol štandardizovaný v rovnakom vydaní (3GPP Release 13), s rovnakým zámerom zvýšiť

energetickú účinnosť a dosah oproti plnohodnotnému LTE (Cat 4). Používa však väčšiu šírku pásma 1,4MHz, čo umožňuje výrazne vyššiu prenosovú rýchlosť (až do 1Mb/s). To ho robí najrýchlejšou v súčasnosti dostupnou LPWAN technológiou. Ako jediná zároveň umožňuje vytvorenie hovoru a prenos hlasu – pomocou protokolu VoLTE, a tiež SMS správ. V porovnaní s NB-IoT je možná plne duplexná komunikácia v jednom čase, latencia dosahuje výrazne nižšie hodnoty (10-15ms), a vďaka podpore dynamického prepínania medzi bunkami je tiež možné použitie v mobilných (pohyblivých) zariadeniach (udávaná je funkčnosť do rýchlosti až 300km/h).

LTE-M dokáže fungovať v rovnakom pásme súčasne s ďalšími LTE štandardmi, pričom nie je potrebný dedikovaný hardvér BTS, iba softvérová podpora. Zariadenia v sieti taktiež komunikujú priamo s internetom a majú vlastné IP adresy. Veľkosť jednej správy nie je obmedzená a je podporované použitie protokolov TCP aj UDP. Pri prenose sa používa modulácia 16QAM v spolupráci s metódami viacnásobného prístupu FDMA pre uplink a OFDMA pre downlink.

Štandard definuje 2 tzv. CE módy fungovania (Coverage Enhancement Modes), ktoré poskytujú rôzne pokrytie – základný Mode A je určená pre vonkajšie pokrytie a je k dispozícii vždy, a Mode B, ktorý je určený pre pokrytie vnútorných priestorov, ale jeho implementácia nie je povinná. Hlbšie pokrytie v móde CE B sa nedosahuje zmenou rádiových parametrov ale inou metódou opakovania v dátových aj riadiacich kanáloch, ale spotreba energie počas vysielania môže byť až o desiatky percent vyššia. [15]

Hlavné výhody sú podobné ako pri NB-IoT:







- Možná dostupnosť všade tam, kde je už dostupné LTE pripojenie bez potreby budovať samostatnú sieť
- Vysoká prenosová rýchlosť
- Vďaka licencovanému pásmu neobmedzený počet správ za deň

Daňou je ale výrazne vyššia spotreba energie, ktorá je tiež ťažko predikovateľná. Niektoré zdroje dokonca nezaradujú LTE-M medzi Low-Power technológie. Druhou nevýhodou je vyššia cena komunikačných modulov a pravdepodobne aj používania siete, účtovaná podľa objemu prenášaných dát. Vzhľadom na tieto vlastnosti je využitie LTE-M vhodné v aplikáciách, kde je potrebný prenos väčšieho množstva údajov a zároveň vysoká spoľahlivosť. Zaujímavým príkladom môžu byť hlásiče nezvyčajných situácií, kedy

zariadenie dlhodobo spí, no v prípade poplachu dokáže urobiť fotografiu a preniesť ju do internetu, alebo vytočiť telefónny hovor. Tiež môže byť vhodné pre platobné brány, sledovanie vozidiel, ovládanie zariadení alebo pre nositeľné zariadenia.

Roaming je rovnako ako pri NB-IoT možný ak majú operátori uzavreté dohody.

V rámci štandardu 3GPP Release 14 bola uvedená aj vylepšená verzia LTE Cat M2, ktorá rozširuje šírku prenosového pásma na 5MHz, čo zvyšuje prenosovú rýchlosť až na (maximálne) 7Mbit/s pri uplinku a 4Mbit/s pri downlinku. [38]

	 Mobility	 Data rates	 Latency	 Indoor penetrations	 Battery life	 Additional services
LTE-M	Full	1/1 Mbps	Low (<1s)	Good	Very long	SMS support
NB-IoT	Limited	250/230 kbps	High (<10s)	Maximum	Maximum	No voice/SMS

Obr. 3.2: Porovnanie vlastností LTE-M a NB-IoT

LTE-M je v súčasnej dobe rozšírený prevažne v Amerike, kde bol prvotne spúšťaný (NB-IoT bol uvádzaný prvotne pre Európsky trh). Na Slovensku prevádzkuje LTE-M operátor Orange [14].

3.1.2 Ultra-Narrowband (Sigfox)

(UNB, ultra-úzkopásmové) Kľúčovou vlastnosťou sú veľmi nízke prenosové rýchlosti, spôsobené moduláciou do extrémne úzkeho prenosového pásma. Tým sa dosiahne nízka hladina akumulovaného šumu, teda je možná vysoká citlivosť pri prijímaní, a tým veľmi vysoký dosah. Problémom tohto prístupu je, že veľmi nízke prenosové rýchlosti znamenajú dlhý čas vysielania, čo znamená hlavne v nelicencovanom frekvenčnom spektre vysokú náchylnosť na kolízie s vysielaním iných zariadení. Keďže vysielania jednotlivých zariadení nie sú nijak synchronizované a v rovnakom spektre môžu fungovať aj rôzne ďalšie systémy, je nutné zabezpečiť nejaké opatrenia, napríklad odvyselať rovnakú správu viac ráz. To ale zvyšuje energetickú náročnosť a ešte viac zahľucuje éter. Povolená doba vysielania zariadenia (tzv. pracovný cyklus, duty cycle) vo

voľnom frekvenčnom spektre môže byť navyše legislatívne regulovaná, čo je aj prípad Európskej únie. Európsky inštitút pre telekomunikačné normy (ETSI) stanovuje pre zariadenia pracujúce vo voľnom ISM frekvenčnom spektre 868MHz limit pracovného cyklu vo výške 1%, čo znamená, že zariadenie môže v tomto pásme vysielat' maximálne 1% času (864 sekúnd denne) [16]. V USA určuje Federálna komisia pre komunikáciu (FCC) limit 400 milisekúnd na dĺžku jedného súvislého vysielania [17]. Ako následok týchto obmedzení je nutné výrazne limitovat' maximálnu veľkosť správy, určiť limity na množstvo odvysielaných správ denne a prípadne ponúknuť iný dizajn/nastavenie siete v rôznych oblastiach, zohľadňujúci vždy miestne regulácie. Správnym premyslením kompresie správy a časovania odosielania je však možné stále prenášať dostatočné množstvo údajov. [12]

Hlavný predstaviteľom technológií využívajúcich princíp UNB je Sigfox.

3.1.2.1 Sigfox

Táto technológia je rozobratá samostatne podrobne v podkapitole 3.4 z dôvodu, že sme sa ju rozhodli použiť ako primárnu komunikačnú technológiu pre praktickú časť práce. Jeho kľúčovými vlastnosťami sú veľmi vysoký dosah (bežne >20km), minimálna energetická náročnosť a globálna centralizovaná sieť, nevýhodami sú ale veľmi nízka prenosová rýchlosť (100bit/s v EÚ), dĺžka payloadu obmedzená na 12B a limitovaný počet odvysielaných správ za deň (140). Na Slovensku prevádzkuje Sigfox sieť operátor Sigfox Slovakia.

3.1.3 Spread Spectrum (LoRa)

Využívajú techniku rozprestretia spektra, čo je alternatívny spôsob ako dosiahnuť zvýšenie dosahu, bez toho aby bolo potrebné výrazne znížiť prenosovú rýchlosť. Moduláciou prenosu do širokého prenosového spektra vzniká signál s malou amplitúdou podobný šumu, ktorý je veľmi ťažko rušiteľný alebo odpočúvateľný (technológia rozprestretia spektra sa preto dlho využívala výhradne pre armádne účely). Najčastejšie sa používa technika CSS (Chirp Spread Spectrum), ktorá používa na odlíšenie signálu od šumového pozadia princíp tzv. cvrlikavých impulzov podobných zvuku cvrlikania, ktorého prirodzená povaha je v čase lineárne stúpajúca sínusová frekvencia.

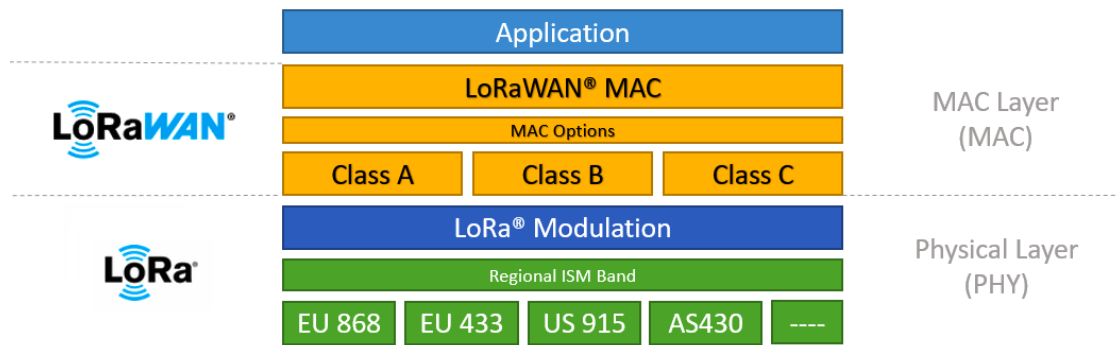
Spektrálna účinnosť tohto riešenia je ale nízka a podstatnou nevýhodou je zlá znášateľnosť iných systémov používajúcich rozprestretie spektra v rovnakom frekvenčnom pásme alebo veľkého množstva zariadení v sieti. V prípadoch, kedy viacero či základňových staníc alebo koncových zariadení na blízkom území vysiela na rovnakých kanáloch, nastáva prekryvanie vysielaní vo frekvenčnom spektre. To môže mať za následok neschopnosť prijímača demodulovať signál a k strate prenášaných dát. Zvyšovaním faktoru rozprestretia sa dosahuje vyššieho dosahu, za cenu nižšej prenosovej rýchlosti a dlhšieho času vysielania. Pri požadovaných dlhých vzdialenostiach prenosu je teda nutné použiť čo najväčší faktor rozprestretia, čo situáciu ešte viac zhoršuje a pravdepodobnosť kolízií stúpa. V nelicencovaných pásmach, kde vysielania nie sú synchronizované môže to viesť k obmedzeniu celkovej prenosovej kapacity siete. Možným riešením je používanie rôznych faktorov rozprestretia a širokých pásiem, no to vyžaduje komplikovanejší manažment siete. [12]

Princíp rozprestretia spektra používa napríklad technológia LoRa.

3.1.3.1 LoRa

LoRa (zo slov Long Range) je dnes pravdepodobne najrozšírenejšou LPWAN technológiou. Jej autorom je francúzska spoločnosť Cycleo, ešte počas vývoja odkúpená korporáciou Semtech, a na trh sa dostala v roku 2012, čo ju robí zároveň jednou z najstarších. Umožňuje obojsmernú komunikáciu na rýchlostiach od 300bit/s do 5,5kbit/s, na veľké vzdialenosti, ktorá prebieha spravidla v nelicencovaných ISM pásmach. Zároveň existuje aliancia s názvom LoRa Alliance, združujúca spoločnosti, ktoré sa podieľajú na ďalšom vývoji a rozširovaní technológie, a ktorej členmi sú aj svetoví hráči ako IBM, MicroChip, Orange alebo Cisco.

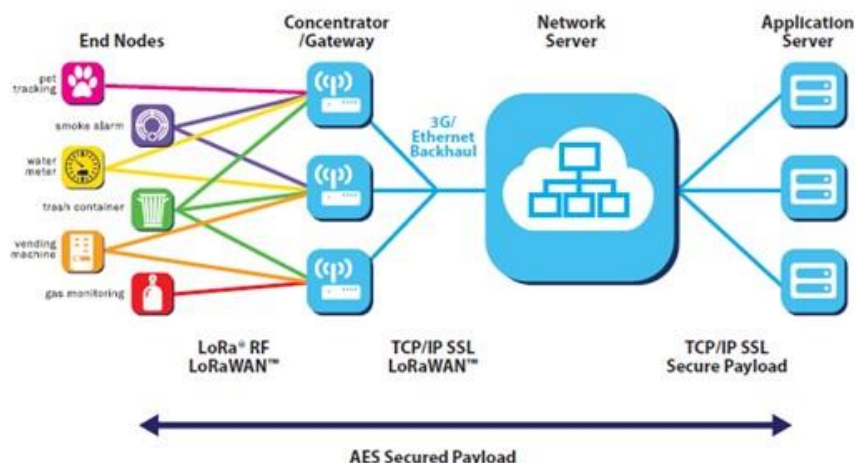
Technológia sa skladá z dvoch častí – LoRa (alebo tiež LoRaPHY) je názov pre samotnú prenosovú technológiu na fyzickej vrstve vrátane patentovanej modulácie, a tá je doplnená vrstvou s názvom LoRaWAN, ktorá definuje sieťový protokol, architektúru siete a ďalšie časti linkovej vrstvy OSI modelu.



Obr. 3.3: Rozdelenie technológie LoRa vo vrstvách

LoRa používa patentovanú moduláciu odvodenú od CSS, čiže techniku rozprestretia spektra. Vďaka tomu je vysoko odolná proti rušeniu a doplerovmu efektu a dosah signálu môže byť oveľa vyšší ako pri celulárnych technológiách. V exteriéri dosahuje okolo 10-15km, no pri ideálnych podmienkach môže dosahovať aj podobné hodnoty ako UNB technológie – rekord je 702km medzi Varšavou a hranicou Holandska v roku 2017 (vysielač sa však nachádzal v balóne vo výške 40km) [18]. Použitie modulácie CSS umožňuje tiež nižšiu výrobnú cenu komunikačných modulov, ktoré nemusia používať vysoko presné súčiastky – frekvenčná odchýlka prijímaného signálu môže byť až do 20% šírky pásma, bez vplyvu na schopnosť dekódovať signál. Tiež je možné používať jednoduchšie algoritmy, čo znižuje spotrebu energie.

Sieťová architektúra LoRaWAN používa viacnásobnú hviezdicovú topológiu (Stars of Stars), kde je brána (základňová stanica) mostom medzi koncovými zariadeniami a centrálnym sieťovým serverom v backende. Brány sú pripojené na sieťový server cez štandardné IP pripojenie, zatiaľ čo koncové zariadenia používajú single-hop bezdrôtovú komunikáciu na bránu/y. Sieťový server riadi sieť, filtruje duplicitné správy (ak rovnaké vysielanie zachytí viacero brán), kontroluje zabezpečenie a preposiela správy cieľovým aplikačným adresátom.



Obr. 3.4: Architektúra systému LoRa

LoRaWAN delí používané spektrum na kanály podľa dostupnej šírky pásma. V Európe je to 10 kanálov – 9 pre uplink – 8 z nich o šírke 125kHz používajúcich LoRa moduláciu (prenosová rýchlosť 250bit/s – 5,5kbit/s), každý aj s možnosťou downlinku + 1 vysokorýchlostný, ktorý používa GFSK moduláciu (do 50kbit/s) – a 1 výhradne pre downlink. 8 ďalších kanálov je vyhradených pre budúce použitie [21].

Vďaka použitej technike CSS sa viacero vysielaní na rovnakej frekvencii, no s rôznym faktorom rozprestretia navzájom neruší, čo umožňuje vytvoriť "virtuálne kanály“, ktoré zvyšujú kapacitu brány. Každý zo štandardných 125kHz kanálov sa takto delí na 6 subkanálov s rôznym použitým faktorom rozprestretia a postupne stúpajúcou prenosovou rýchlosťou. Jeden z nich má navyše druhú skupinu 6 subkanálov so šírkou pásma 250kHz. Priradzovanie koncových zariadení do jednotlivých subkanálov a ich rádiovú konfiguráciu dynamicky manažuje sieťový server, čo umožňuje aj podporu adaptívnej zmeny prenosovej rýchlosti (ADR). Výber použitého subkanálu tvorí vždy kompromis medzi dosahom a trvaním prenosu.

Dĺžka payloadu uplink správy môže byť maximálne 64-256 bajtov v závislosti od najvyššieho povoleného faktoru rozprestretia, navyše ďalších minimálne 13B obsadzuje protokol LoRaWAN [19]. Pri móde najširšieho rozprestretia je možná dĺžka len 51 bajtov. Maximálny počet správ, ktoré je možné poslať denne nie je LoRaWAN-om obmedzený. Keďže ale vysielanie prebieha v nelicencovaných pásmach, je na prevádzkovateľovi systému, aby dodržal legislatívne regulácie platné v danej oblasti. Tieto obmedzenia sú bližšie popísané v podkapitole 3.4.1, no pre Európsku úniu ide najmä o 1% pracovný

cyklus vysielania jedného zariadenia. Počet správ, ktoré je potom možné odvyselať závisí od použitých prenosových rýchlostí.

LoRaWAN definuje 3 triedy koncových zariadení, respektíve módy v ktorých môžu pracovať, na základe spotreby energie a dostupnosti downlink komunikácie:

- Trieda A – je určená najmä pre batériovo napájané senzory, musia ju podporovať všetky zariadenia. Každý uplink prenos je nasledovaný dvoma krátkymi oknami pre príjem downlink komunikácie. Doručenie zostupnej správy v inom okamihu nie je možné a musí počkať do najbližšieho uplink vysielania.
- Trieda B – je určená najmä pre batériovo napájané akčné členy. Zariadenia môžu pre zostupnú komunikáciu nastaviť plánované extra prijímacie okno. Pre synchronizáciu so serverom sa používajú beacon správy vysielané bránou.
- Trieda C – je určená najmä externe napájaným akčným členom. Zariadenia majú zariadenia otvorené prijímacie okná nonstop, okrem vlastného vysielania, zostupnú správu je teda možné doručiť kedykoľvek.

V prípade že je sieťový server spravovaný prevádzkovateľom aplikácie, výhodou je plná kontrola nad celou sieťou a jej konfiguráciou. LoRa zároveň explicitne neurčuje použité rádiové pásmo, čiže je možné ju v prípade potreby (a dostupnosti potrebného hardvéru) prevádzkovať aj na iných ako štandardných frekvenciách.



Obr. 3.5: Outdoorový LoRa gateway od spoločnosti Friendcom

Obr. 3.6: Vývojový LoRa komunikačný modul

Zabezpečenie

LoRaWAN využíva 2 vrstvy zabezpečenia – jednu pre sieť a druhá pre aplikáciu.

Vrstva zabezpečenie siete zaisťuje autenticitu uzlov v sieti – každé koncové zariadenie je identifikované unikátnym globálnym identifikátorom založeným na EUI-64 nazývaným DevEUI, a siete sú identifikované 24-bitovým globálne unikátnym identifikátorom prideleným LoRa alianciou. Vrstva zabezpečuje integritu správ a ochranu pred replay útokmi pomocou počítadiel rámcov.

Druhá vrstva zabezpečuje end-to-end šifrovanie pre payloady, od koncových zariadení až po aplikačný server, používa sa šifrovanie AES. Pokiaľ ide o komunitnú sieť, jej prevádzkovateľ teda nemá prístup k prenášaným dátam koncového užívateľa. Každé koncové zariadenie vlastní vlastný 128-bitový kľúč nazývaný AppKey, pomocou ktorého sa odvodzujú šifrovacie kľúče.

LoRaWAN podporuje dve metódy pripojenia koncového zariadenia k sieti:

- OTAA (Over The Air Activation) – “Aktivácia vzduchom“
- ABP (Activation By Personalization) – “Aktivácia pomocou personalizácie“

Pri použití metódy OTAA vyšle zariadenie žiadosť o pripojenie do siete. Sieť po prijatí tejto požiadavky overí identifikátor zariadenia (AppEUI) a poskytnutý aplikačný kľúč (AppKey). Ak sú AppEUI a AppKey platné, koncovému zariadeniu sa späť odošle potvrdenie o prijatí do siete a sú odvodené dva kľúče relácie použité pre šifrovanie:

- Network Session Key (NwkSKey) – pre ochranu integrity a šifrovanie MAC príkazov
- Application Session Key (AppSKey) – pre end-to-end šifrovanie payloadu

NwkSKey je distribuovaný do LoRaWAN siete na zabezpečenie resp. overenie autenticity a integrity paketov. AppSKey je distribuovaný do aplikačného servera na zabezpečenie šifrovania a dešifrovania payloadu [20].

Pri použití ABP neexistuje žiadna inicializačná komunikácia medzi koncovým zariadením a sieťou. Po pripojení do siete môže zariadenie okamžite začať s odosielaním údajov. Šifrovacie kľúče AppSKey a NwkSKey sú nastavené jednorazovo v koncovom zariadení a na sieťovom serveri, musia byť na oboch stranách rovnaké.

Signály vysielané nízkym výkonom pri modulácii CSS je samé o sebe veľmi náročné detekovať a zachytiť. V kombinácii so vzájomným overovaním a šifrovaním sa dosahuje LoRa garanciu, že sieťová prevádzka [20]:

- nebude pozmenená,
- pochádza z legitímneho zariadenia,
- je nezrozumiteľná v prípade odpočúvania,
- nebude odchytená a pozmenená treťou stranou.

The Things Network

(TTN) Ide o projekt globálnej, otvorenej, komunitnej LoRa siete, ktorej časti sú vlastnené a prevádzkované užívateľmi. Projekt je tiež členom LoRa Alliance. Do TTN sa môže pripojiť akýkoľvek prevádzkovateľ LoRa brán/y, ktorý akceptuje manifest projektu. Komunikácia s TTN prebieha cez bežné internetové pripojenie a servery TTN vedú zastúpiť rolu sieťového servera (z klasickej architektúry LoRa). Vytvorená a spustená bola v roku 2015, pričom v súčasnosti je do nej zapojených vyše 19 000 rôznych brán v 151 krajinách po celom svete, čo ju robí najväčšou IoT sieťou na svete. Najväčšiu hustotu má v Európe. Otázne je ale reálne pokrytie – keďže jednotlivé podsiete môžu mať rôzne konfigurácie a brány rozdielne výkony, dostupnosť je možné uvádzať len približne na úrovni miest a oblastí, polohou jednotlivých brán. Projekt zahŕňa aj rozhranie pre správu koncových zariadení a integráciu s ďalšími cloud platformami (aplikačnými servermi). Komunikácia v celej sieti je zabezpečená rovnako end-to-end šifrovaním AES, ako v lokálnej sieti.

Pre nekomerčné účely je jej používanie zdarma, existuje aj podpora komerčného využitia, ktoré zastrešuje Holandská organizácia s názvom The Things Industries, ktorej zakladajúci členovia stoja za vznikom projektu TTN. Hoci z pohľadu komunikácie je TTN navrhnutá ako decentralizovaná infraštruktúra, spolieha sa na centrálné servery zabezpečujúce správu siete. Taktiež dostupnosť a funkčnosť podsietí a jednotlivých brán je v réžii ich prevádzkovateľov a nemôže byť garantovaná. Pre nekomerčné použitie definuje TTN pravidlá férového používania, ktoré obmedzujú využívanie siete na podobné kapacity ako poskytuje technológia Sigfox². Striktne ide o 10 downlink správ za

² Pri použití kanálov s pomalou prenosovou rýchlosťou a teda vysokým dosahom je to ešte oveľa menej.

deň a priemernú dobu jedného vysielania na 30 sekúnd, po prepočte na denný duty cycle 1% je slovné odporúčanie udržiavať dĺžku payloadu do 12B a interval medzi vysielaním aspoň niekoľko minút [19].

Nevýhodami LoRa sú teoreticky možná nižšia spoľahlivosť v oblastiach s veľkým množstvom zariadení, kedy sa prekročí kapacita siete a môže dochádzať ku kolíziám. Každá správa sa vysielala len raz a nie je možné overiť či bola správa úspešne doručená. Keďže LoRa môže byť prevádzkovaná ako súkromná sieť, ktorých sa môže na jednom území nachádzať viacero, vo veľkých mestách môže tento fakt (najmä v budúcnosti) predstavovať problém. Ďalšou nevýhodou je mierne vyššia cena obstarávacia cena komunikačných modulov a modemov, keďže spoločnosť Semtech pôsobí ako ich jediný výrobca. Potenciálne najväčšou nevýhodou však môže byť nutnosť prevádzkovať a spravovať vlastnú sieť brán a centrálného sieťového servera. Tento problém odpadá pri využití infraštruktúry projektu The Things Network, kde ale nie je garantovaná dostupnosť služby, alebo pri využívaní komerčnej siete, ktorú prevádzkuje spoločnosť tretej strany, väčšinou telekomunikačný operátor.

Na Slovensku prevádzkujú verejné siete operátori Orange, Slovanet a Antik [14], pričom je možné využiť aj komunitné siete zapojené do projektu The Things Network.

3.1.4 Telegram Splitting

Je najnovším prístupom pri technológiách LPWAN, zároveň je jediným zatiaľ štandardizovaným LPWAN riešením pre nelicencované pásma. Základný princíp spočíva v rozdelení dátového paketu na množstvo menších sub-paketov, ktoré sú odvysielané v rôznych časoch na rôznych frekvenciách. Modulácia prebieha do UNB pásma. Vzhľadom na krátku dĺžku jednotlivých sub-paketov je ich vysielací čas len niekoľko milisekúnd a teda princíp Telegram Splitting môže ťažiť z vysokej spektrálnej efektívnosti UNB technológií a zároveň prekonávať ich nevýhody vyplývajúce z dlhého času vysielania. Minimalizuje sa pravdepodobnosť kolízií, a v prípade že by nejaké predsa len nastali, je možné pri spätnej rekonštrukcii správy malé množstvo stratených dát jednoducho dopočítať použitím samoopravného kódu (FEC). Kombináciou vlastností vzniká vysoká odolnosť voči doplerovmu efektu a tým pádom možnosť použiť technológiu pre rýchlo

pohybujúce sa objekty. Následkom rozdelenia správy na množstvo sub-paketov vysielaných časovo aj frekvenčne oddelene je tiež prirodzene vysoká bezpečnosť – je prakticky nemožné prenos odpočúvať bez znalosti algoritmu rozdelenia správy. Princíp Telegram Splitting je štandardizovaný organizáciou ETSI (štandard TS 103 357 z roku 2018), čo zabezpečuje jeho nezávislosť od dodávateľa a dlhodobú interoperabilitu [12].

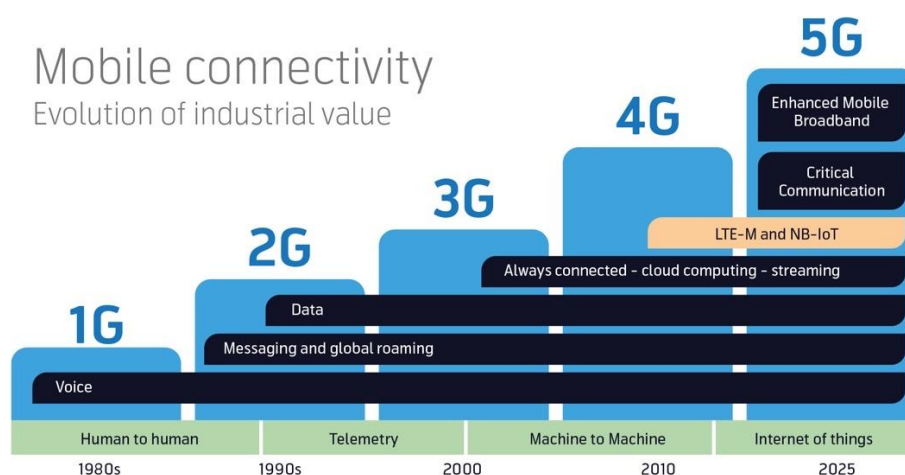
Jedinou technológiou využívajúcou princíp Telegram splitting je zatiaľ MlIoT, oficiálne predstavený len v roku 2020. Ide o patentovanú technológiu nemeckej spoločnosti BehrTech. Zatiaľ nie sú verejne dostupné žiadne siete a v súčasnej dobe prebieha hlavne vývoj modemov a komunikačných modulov.

Okrem popísaných existuje množstvo ďalších LPWAN technológií a štandardov, spomenúť možno Weightless, nWave, DASH7, IQRf, NB-Fi.

3.2 Širokopásmové mobilné technológie

Pod širokopásmovými mobilnými technológiami rozumieme bežné bunkové siete vychádzajúce zo štandardu GSM, ktoré poznáme z mobilných telefónov. V súčasnosti sa nachádzame na prelome 4-tej a 5-tej generácie týchto sietí. Treba zdôrazniť, že jednotlivé generácie sietí nepredstavujú konkrétne technológie, ale ideologovú skupinu technológií z rovnakého obdobia, pričom ide prevažne o marketingovú záležitosť. Prvá generácia (1G) predstavovala analógovú sieť a technológie. Druhá generácia (2G) je už digitálna a zavádza textové správy, roaming a prvé dátové služby. Tretia generácia (3G) rozširuje možnosti prenosu dát a značne zvyšuje prenosové rýchlosti. Štvrtá generácia (4G) prináša ďalšie výrazné zvýšenie rýchlostí a prináša nové technológie pre M2M komunikáciu. Piata generácia (5G) je práve v priebehu vývoja a nasadzovania a okrem markantného zvýšenia kapacity siete a rýchlostí prinesie mnohé nové koncepty a možnosti pre oblasť IoT. V ďalšom texte sa budeme venovať len generáciám od druhej vyššie. Výhodou a zároveň nevýhodou týchto technológií je, že zariadenia sú priamo pripojené do internetu, teda majú vlastnú IP adresu pomocou ktorej je ich možné priamo kontaktovať. Štandardy mobilných sietí sú celosvetovo široko adoptované, a teda dostupnosť sietí a modemov a ich ceny sú na výhodnej úrovni. Oproti iným IoT technológiám majú tiež natívnu podporu pre prenos hlasu, SMS správ a roaming, čo môže byť v niektorých

aplikáciách užitočné. Spoločnými nevýhodami sú ale vysoká spotreba energie, potreba kontraktu s operátorom siete, vyššie prevádzkové náklady a potreba riešiť veľké množstvo frekvenčných pásiem, v ktorých sú jednotlivé siete v jednotlivých krajinách prevádzkované. V prípade potreby medzinárodného roamingu je potrebné, aby modem podporoval frekvenčné pásma použité všetkými sieťami, v ktorých má fungovať.



Obr. 3.7: Časový vývoj generácií mobilných sietí

3.2.1 2G a 4G

V minulosti (“pred-IoT ére“) sa na komunikáciu typu M2M používala sieť 2G, založená prevažne na základnej technológii GSM a nadstavbách GPRS (General Packet Radio Service) a EDGE (Enhanced Data rates for Global Evolution), ktoré poskytujú možnosti dátových prenosov do rýchlosti približne 480kbit/s. Kvôli svojej jednoduchosti, relatívne nízkej energetickej náročnosti, nízkej cene a dobrému dosahu, ktorý môže dosahovať vzdialenosť do 35km od základňovej stanice je GSM v oblasti IoT ešte stále aj dnes využívaným riešením konkurujúcim moderným prenosovým technológiám. V súčasnosti ale vo svete už prebieha postupné ukončovanie prevádzkovania 2G sietí, preto nie je rozumné vyvíjať nové riešenia na nich založené. V Európe dôjde najprv k vypnutiu sietí 3G, až neskôr 2G, kvôli zachovaniu konektivity pre staré mobilné telefóny. V Amerike však zvolili opačný postup a 2G siete budú vypnuté ako prvé, v mnohých krajinách Ázie už vypnuté boli. Riešenie ponúka technológia EC-GSM-IoT, čo je samostatná, no s GSM spätne kompatibilná LPWAN technológia optimalizovaná pre vysoký dosah a nízku spotrebu, ktorá môže byť prevádzkovaná súčasne so sieťami

nových generácií na rovnakom hardvéri BTS staníc, len s pomocou softvérového rozšírenia [22].

Alternatívou, ktorá zároveň poskytuje vysoké prenosové rýchlosti sú siete 4G, založené prevažne na technológiách LTE (Long Term Evolution). LTE je však súbor mnohých štandardov, preto už treba rozlišovať, či požadujeme vysokorýchlostné širokopásmové, alebo nízkorýchlostné úzkopásmové pripojenie. V druhom prípade sú dostupné napríklad technológie NB-IoT a LTE-M, ktoré už boli rozoberané v podkapitole 3.1. V prípade potreby vysokorýchlostného pripojenia sú k dispozícii technológie LTE-A (LTE-Advanced) a LTE-A Pro, ktoré umožňujú teoretické prenosové rýchlosti až do 3Gbit/s. V prípade aplikácií, kde je potrebné prenášať multimediálne dáta alebo veľké súbory teda dokážu poskytnúť plnohodnotnú vysokorýchlostnú konektivitu na internet. To sa ale odzrkadľuje v značne vyššej cene modemov, poplatkov za prenos dát a vysokej energetickej náročnosti.

3.2.2 5G

5G taktiež nie je jedna konkrétna technológia, ale (bude to) súbor množstva štandardov s rozdielnymi cieľmi. Všeobecné predpoklady, ktoré sa od 5G očakávajú sú použitie (aj) milimetrových rádiových vln (frekvencie do 100GHz), rozdeľovanie oblastí na menšie bunky (kompenzuje menší dosah a horšie šírenie vysokých frekvencií a zároveň umožňuje zväčšenie počtu pripojených zariadení na rovnakej ploche), používanie mnohoanténnych systémov (MIMO, umožní komunikáciu na viacerých pásmach súčasne), tvarovanie vyžarovacieho diagramu (použitím MIMO a sofistikovanou moduláciou/spracovaním signálu, umožní komunikáciu s nižšou spotrebou) a plne duplexná komunikácia priamo na rádiovéj úrovni [23]. Ďalším prínosom má byť čiastočná decentralizácia siete, respektíve umožnenie priamej komunikácie medzi zariadeniami s vynechaním operátora, čo umožní vyššiu rýchlosť a nižšiu latenciu. Zároveň sa predpokladá že 5G siete budú dokázať pripojiť na ploche 1km² až 1 milión zariadení. Samozrejme ale treba rozlišovať bunky v mestách, špeciálne v husto frekventovaných častiach miest, a bunky pokrývajúce veľké mimomestské oblasti – s čím ráta aj využitie rôznych frekvenčných spektier pre rôzne použitia. 5G všeobecne počíta s využívaním eSIM kariet (embedded-SIM), ktoré neobmedzia veľkosť zariadení a poskytnú možnosť spravovať SIM na diaľku.

Medzinárodná telekomunikačná únia (ITU) definuje pre 5G rozdelenie na tri hlavné spôsoby použitia:

- eMBB (enhanced Mobile BroadBand) – vysokorychlostnú mobilnú komunikáciu
- URLLC (Ultra Reliable Low Latency Communications) – vysoko spoľahlivú komunikáciu s nízkou latenciou
- mMTC (massive Machine Type Communications) – hromadnú komunikáciu medzi strojmi (M2M)

V súčasnosti stále prebieha vývoj a štandardizácia technológií pre 5G, ktoré sú vydávané konzorciom 3GPP priebežne. Zatiaľ bola v roku 2018 vydaná prvá “fáza“ štandardov (Release 15), ktorá položila základy architektúry 5G sietí a definovala základnú komunikačnú technológiu s názvom 5G NR (New Radio); a v roku 2020 druhá fáza (Release 16), ktorá definuje základy množstva ďalších funkcionalít ktoré sa v kontexte 5G spomínajú, napríklad: komunikačné technológie v pásmach 50-100GHz, interoperabilitu so satelitnými systémami, presnú lokalizáciu zariadení, funkcie pre IoT a IIoT aplikácie, vylepšenia pre NB-IoT a LTE-M, podporu viacerých SIM kariet, LAN siete postavené na 5G, špecifiká pre mission-critical aplikácie, URLLC komunikáciu, vylepšenia pre V2X komunikáciu, edge-computing, atď. Na rok 2022 je plánované dokončenie 3. fázy (Release 17), ktorá má dokončiť predchádzajúco menované funkcionality. Počíta sa aj s ďalšími fázami, v súčasnosti sa začína s určovaním cieľov pre Release 18. [24]

5G mobilné siete, ktoré sú nasadzované v súčasnosti sú postavené na technológií 5G NR (New Radio). Ide o komunikačnú technológiu určenú pre vysokorychlostné pripojenie mobilných zariadení (rámec eMBB), ktorá sľubuje oproti predchádzajúcej generácii približne desaťnásobné zvýšenie prenosovej rýchlosti (do 20Gbit/s) a zníženie latencie pripojenia na jednotky milisekúnd.

Pre oblasť IoT sa v 5G počíta (aj) s využívaním už existujúcich technológií NB-IoT a LTE-M, pre ktoré ale pribudnú rôzne vylepšenia a nové funkcionality. Tieto štandardy sú ale ešte vo vývoji a širšie realizovanie myšlienok, ktoré 5G pre svet IoT ponúka možno očakávať najskôr po dokončení vydania Release 17. Zaujímavé možnosti môžu poskytnúť tiež satelitné 5G siete, ktoré zabezpečia dostupnosť prakticky kdekoľvek na svete. Tieto technológie ale pravdepodobne nebudú navrhnuté organizáciou 3GPP, tá definuje len spôsoby prepojenia a kooperácie 3GPP technológií s technológiami tretích

strán [35]. Prvým v súčasnosti realizovaným projektom ktorý do tejto skupiny môže spadať je systém Starlink od spoločnosti SpaceX. Ten síce vyžaduje použitie malých parabolických antén, technicky je ale možné komunikovať so satelitmi na nízkej obežnej dráhe aj s použitím bežných prúťových antén.

Zavádzanie 5G technológií je teda postupný proces, ktorý bude trvať ešte niekoľko rokov. V súčasnosti situáciu tiež komplikujú konkurenčné a politické spory okolo výrobcov 5G technológií (blokované spoločnosti Huawei) a tiež kritický nedostatok 5G modemov a čipov všeobecne (rok 2021). Kým pred rokom bolo rýchle nasadzovanie 5G sietí na hlavných svetových trhoch prioritou všetkých veľkých operátorov, aktuálne okolnosti značne schladili očakávania a spomalili postup.

3.3 Lokálne pripojenie WLAN

Za vhodnú alternatívu môžeme považovať aj lokálne siete WLAN. Túto oblasť pokrýva najmä súbor štandardov 802.11 organizácie IEEE, s marketingovým názvom Wi-Fi. Hoci je tento pojem zaužívaný pre pripojenie spotrebiteľských zariadení k internetu, a to v rámci budovy, súbor 802.11 v súčasnosti zabaľuje 45 samostatných existujúcich alebo pripravovaných štandardov [25]. Množstvo z nich sú len doplnky alebo rozšírenia iných, ale nachádza sa medzi nimi množstvo samostatných prenosových technológií pre komunikáciu rôznych typov zariadení v rôznych frekvenčných pásmach. Spoločnou výhodou WLAN technológií je, že nie je potrebný externý operátor, čím odpadá časť prevádzkových nákladov aplikácie na prístup do siete.

3.3.1 802.11b/g/n

Protokoly 802.11b, 802.11g a 802.11n sú práve tými, pod ktorými sa bežne chápe pojem Wi-Fi. Všetky pracujú v nelicencovanom frekvenčnom pásme 2,4GHz.

Štandard 802.11n (2009) poskytuje dobrý kompromis medzi vlastnosťami, želanými pri štandardnom domácom použití a tými, ktoré vyžadujú aplikácie IoT. Pri použití 2,4GHz frekvenčného pásma (802.11n podporuje aj 5GHz pásmo) je dosah siete v exteriéri okolo 100 metrov, čo môže byť často dostatočné. Šírka kanála môže byť 20/40MHz, pre zabránenie kolíziám sa používa technika CSMA-CA (CSMA Collision

Avoidance). Maximálna prenosová rýchlosť je obmedzená počtom použitých antén a MIMO konfigurácie, čo umožňuje škálovať komunikačný hardvér zariadenia s prihliadaním na cenu, v závislosti od požiadaviek. Maximálna možná teoretická rýchlosť je 600Mbit/s. Veľkou výhodou je dostupnosť nespočetného množstva modemov a rôznych komunikačných modulov, kombinovaných aj s inými technológiami, za nízke ceny. Spätne kompatibilné štandardy 802.11g a najstarší 802.11b je možné použiť v prípade špecifických požiadaviek na jednoduchý hardvér alebo pomalšie, no stabilnejšie spojenie.

3.3.2 802.11ah a 802.11af

802.11ah (Wi-Fi HaLow, 2017) je štandard pre komunikáciu v Sub-GHz pásme, prioritne určený pre použitie v IoT a M2M komunikáciu. Pracuje na nelicencovaných kanáloch v pásme 900MHz, čo umožňuje dosah vo voľnom priestore až do 1km. Šírka prenosového pásma je 1-16MHz, prenosová rýchlosť sa môže pohybovať v rozsahu od 150kbit/s do 40Mbit/s v závislosti od použitej kódovej schémy. Zabezpečenie prebieha pomocou rovnakých protokolov WPA, aké používajú štandardy pre počítače. Technológia je technicky založená na vysokorýchlostnom štandarde 802.11ac, no implementuje viaceré zmeny na fyzickej a linkovej vrstve.

Výhodou je nízka energetická náročnosť, ktorá môže pri nízkych dátových prenosoch konkurovať LPWAN technológiám. Sú definované 2 režimy správy energie – aktívny režim a úsporný režim. AP periodicky zasiela beacon rámec, ktorý obsahuje informáciu o tom, pre aké stanice má nové pakety na doručenie. V úspornom režime je stanica pravidelne prebúdzaná týmto rámcom a pokiaľ pre ňu nie sú dostupné nové pakety, môže prejsť do spánku do obdržania ďalšieho beacon rámca. Prístupový bod rozdeľuje pripojené stanice do skupín, aby sa čo najviac obmedzilo zbytočné zobúdzanie zariadení a tiež rušenie medzi kanálmi.

802.11af je podobná technológia so strednou až vyššou prenosovou rýchlosťou, no prevádzka prebieha v nepoužívaných televíznych rozsahoch (54-790MHz) v danej krajine. [25]

3.3.3 802.11p a 802.11bd

802.11p (WAVE – Wireless Access for the Vehicular Environment, 2010) je štandard pre vzájomnú komunikáciu automobilov a automobilov s okolitou cestnou infraštruktúrou (V2X). Komunikácia prebieha v licencovanom frekvenčnom pásme 5,9 GHz, v kanáloch šírky 10MHz rýchlosťou 3-27Mbit/s, pri vzdialenosti objektov 150 metrov až 1 kilometer. Hoci nepodporuje priamo konektivitu vozidiel k internetu, umožňuje iný typ komunikácie ktorý bude v oblasti smart vozidiel potrebný v moderných aplikáciách.

Štandard 802.11bd (Enhancements for Next Generation V2X) predstavuje pripravovanú nadstavbu nad 802.11p a bude sa zameriavať na komunikáciu V2X modernej generácie, ako napríklad definovanie tzv. groupcast komunikácie v skupine viacerých vozidiel pri autonómnom riadení. Finalizácia sa očakáva v roku 2022. [25]

3.4 Sigfox

Sigfox je jedným z najväčších hráčov na trhu LPWAN technológií. Ide o patentovanú technológiu, ktorej investormi sú spoločnosti ako Intel, Telefónica, Samsung, Eutelsat, či SK Telecom. Jej kľúčovými vlastnosťami sú extrémne nízka prenosová rýchlosť, len 100bit/s (resp. 600bit/s), čo na oplátku ale prináša veľmi nízku energetickú náročnosť, ktorá je veľkou výhodou najmä v batériovo napájaných zariadeniach. Technológia poskytuje vysokú spoľahlivosť prenosu - každá správa je pri štandardnom nastavení vysielaná 3-krát, na mierne odlišných frekvenciách a s niekoľkosekundovým odstupom. Týmto prístupom sa dosiahne značné zvýšenie pravdepodobnosti, že ju z dôvodu väčších vzdialeností alebo vo viac zarušenom prostredí zachytí aspoň nejaká základňová stanica v okolí, respektíve že sa správu podarí úspešne doručiť aj ak pri niektorom z vysielaní nastala kolízia s vysielaním iného zariadenia v okolí, čo by inak nebolo možné odhaliť. Základňová stanica a sieť potvrdenie príjmu správy nepotvrďuje, ale je zodpovedná za riešenie duplicit ktoré z princípu vznikajú prijatím rovnakej správy viacnásobne, väčšinou aj viacerými základňovými stanicami súčasne.

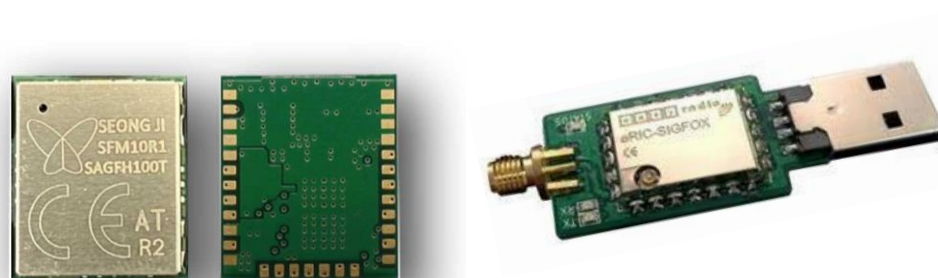
Technológiu od roku 2010 vyvíja a prevádzkuje za pomoci partnerov rovnomená, francúzska spoločnosť Sigfox, ktorá vlastní patenty na jej komunikačné princípy. Siete samotné sú prevádzkované partnerskými spoločnosťami, pričom platí pravidlo, že v

každej krajine prevádzkuje sieť len jeden operátor. Sieť je postavená na single-hop hviezdicovej topológii, kedy každá základňová stanica má (prostredníctvom VPN) priame pripojenie do centrálného Sigfox Cloudu, kde sa vykonáva kontrola integrity, autorizácie, zjednotenie duplicit, metrika, filtrovanie a archivovanie, a odkiaľ sú prostredníctvom callbackov správy ďalej distribuované adresátom. Základňové stanice nekomunikujú navzájom.



Obr. 3.8: Architektúra systému Sigfox

Koncové zariadenie sa do siete pripája prostredníctvom modemu alebo samostatného komunikačného modulu, ktorý má jedinečný 32-bitový UID identifikátor, ktorým sa v sieti identifikuje. Aktivácia prístupu zariadenia do siete prebieha jednorazovo pomocou prideleného PAC kódu, preto nie je nutné použitie SIM kariet. Všetky zariadenia určené na komerčnú prevádzku musia prejsť certifikáciou, čo zabezpečuje ďalšiu úroveň spoľahlivosti technológie. Ich cena na trhu sa pohybuje v rozsahu od 10 do 50eur, v prípade veľkoodberu už od jednotiek eur.

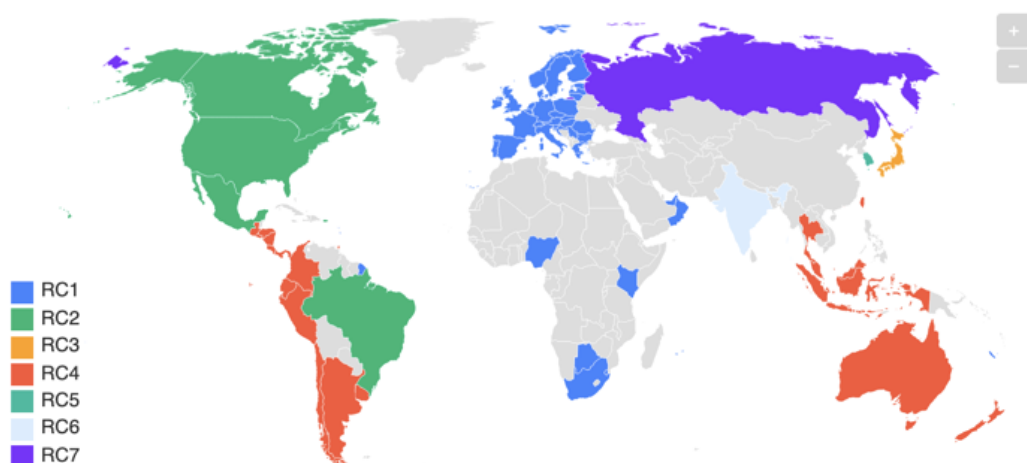


Obr. 3.9: Sigfox modem a vývojársky USB modul

3.4.1 Rádiové konfigurácie

Komunikácia v éteri prebieha vo voľnom nelicencovanom frekvenčnom pásme. Pre Európu je to pásmo 868MHz v Európe, inde sú to pásma v okolí 868MHz a 915MHz, v minulosti sa v Ázii používalo aj pásmo 433MHz. Sigfox delí svet na 7 regiónov, označených RC1-RC7 (RC – Radio Configuration), pričom pre ne platia rozdielne pravidlá a nastavenia siete. Mapu regiónov a konkrétne parametre a špecifiká pre ne sú uvedené nižšie. V tabuľke je kurzívou zvýraznený región RC1, do ktorého spadá Európa.

Sigfox Geographical Availability



Obr. 3.10: Mapa regiónov technológie Sigfox

	<i>RC1</i>	RC2	RC3	RC4	RC5	RC6	RC7
Uplink stredná frekvencia (MHz)	<i>868.130</i>	902.200	923.200	920.800	923.300	865.200	868.800
Downlink stredná frekvencia (MHz)	<i>869.525</i>	905.200	922.200	922.300	922.300	866.300	869.100
Uplink prenos. rýchlosť (bit/s)	<i>100</i>	600	100	600	100	100	100
Downlink prenos. rýchlosť (bit/s)	<i>600</i>	600	600	600	600	600	600
Odporúčané EIRP (dBm)	<i>16</i>	24	16	24	14	16	16
Špecifické požiadavky	<i>Duty cycle 1%</i>	Frekv. skákanie, On-Air-Time	Listen Before Talk	Frekv. skákanie, On-Air-Time	Listen Before Talk	-	Duty cycle 1%

Tab. 3.1: Parametre a špecifiká technológie Sigfox v regiónoch

Z použitia ISM pásiem vyplývajú v rôznych krajinách rôzne legislatívne obmedzenia a požiadavky, ktoré sú uvedené v poslednom riadku tabuľky. Kým funkciu frekvenčného skákania (náhodný výber kanálu pre odvysielanie správy) využíva Sigfox predvolene vo všetkých regiónoch, obmedzenie maximálneho času vysielania vyžaduje použitie vyššej prenosovej rýchlosti. Najzásadnejšie je obmedzenie pracovného cyklu na 1% vysielacieho času – po prepočte s prenosovou rýchlosťou 100bit/s, plnou dĺžkou správy a štandardným opakovaním vysielania jednej správy 3-krát, to vychádza na odvysielanie maximálne 144 správ denne. Sigfox kvôli tomu globálne stanovuje limit pre jedno zariadenie na 140 prenesených správ za deň vo vzostupnom smere, a len 4 správy denne v zostupnom.

Zariadenia môžu byť vytvorené pre jednu alebo viacero rádiových konfigurácií. Väčšina z dostupných komunikačných modulov je navrhnutá pre jednu z nich, no existujú modemy, ktoré implementujú požiadavky na všetky regióny a teda dokážu fungovať v ktoromkoľvek z nich.

Sigfox tiež definuje hodnotenie certifikovaných zariadení/modemov do tzv. tried, na škále od U0 po U3 podľa ich schopnosti doručiť správu v pomere s použitým EIRP. Trieda U0 je najlepšie hodnotenie, na ktoré by mali všetky vyvíjané zariadenia cieľiť. V špecifických prípadoch, keď sa napr. navrhuje systém o ktorom sa vie že bude prevádzkovaný v lokalite s dobrým pokrytím, je možné kvôli dosiahnutiu nižšej energetickej náročnosti alebo výrobnéj ceny cieľiť len na požiadavky nižšej triedy. Tiež je možné EIRP meniť nastavením dodatočne, a to aj na diaľku, prostredníctvom downlink komunikácie. Znížením vyžarovaného výkonu je možné ušetriť až 50% potrebnej energie oproti nastaveniu potrebnému pre splnenie požiadaviek triedy U0.

3.4.2 Sigfox Monarch

Väčšina aplikácií vo svete internetu vecí je predávaná a prevádzkovaná lokálne. Sú však prípady, kedy je potrebná funkčnosť naprieč viacerými regiónmi. Aby sa zariadenie, ktoré podporuje viacero rádiových konfigurácií vedelo automaticky nastaviť na potrebnú z nich v mieste kde sa aktuálne nachádza, Sigfox sieť ponúka funkciu siete s názvom Monarch. Ide o vysielanie špeciálnych beacon správ niektorými vybranými základňovými stanicami, ktoré obsahujú informáciu o použitej rádiovkej konfigurácii v danej oblasti. Modem v koncovom zariadení musí byť schopný tieto správy detekovať,

no systém je navrhnutý tak, aby dodatočná podpora tejto funkcie nemala výrazný vplyv na spotrebu energie alebo komplikovanosť prijímača (Pre beacon správy je použitá veľmi jednoduchá OOK – On-Off keying amplitúdová modulácia). Takto vybavené koncové zariadenie potom môže prechádzať medzi ľubovoľnými RC regiónmi a komunikačný modul sa pri zachytení prvého Monarch beaconu automaticky prispôbi miestnej rádiovkej konfigurácii. Keďže sieť Sigfoxu je globálne jednotná, nie je potrebné žiadne preregistrovanie zariadenia ani na logickej úrovni a platia preň celosvetovo rovnaké kontrakty, tarifikácie a podobne. Vďaka tejto funkcii teda Sigfox umožňuje plne automatický celosvetový roaming.

3.4.3 Dátový formát a komunikácia

Maximálna dĺžka obsahu jednej správy (payload) vo vzostupnom smere (uplink, teda od vysielateľa na základňovú stanicu) je 12 bajtov, pričom je možné poslať aj prázdnu správu. Sigfox protokol ale určuje niekoľko preddefinovaných formátov vysielaného rámca, pre dĺžky payloadu 0, 1, 4, 8, alebo 12 bajtov. Pokiaľ je vstupná dĺžka payloadu iná, firmvér modemu ho automaticky doplní prázdny bajtmi podľa nasledujúcej schémy:

Bytes of data	0	1	2	3	4	5	6	7	8	9	10	11	12
Payload size in bytes	0	1	4		8				12				

Dĺžka payloadu má priamy vplyv na dĺžku vysielania daného rámca. V závislosti od použitej prenosovej rýchlosti (podľa regiónu) sú potom časy nasledovné:

Dĺžka payloadu	Čas vysielania rámca	
	RC1/RC3/RC5-7	RC2/RC4
< 1 bit	1,1 sekundy	190 ms
2 bity - 1 bajt	1,2 sekundy	200 ms
2 - 4 bajty	1,45 sekundy	250 ms
5 - 8 bajtov	1,75 sekundy	300 ms
9 - 12 bajtov	2 sekundy	350 ms

Tab. 3.2: Dĺžka vysielania pri rôznych dĺžkach správy

Hlavička a pätička rámca tvorí pri uplinku ďalších 14-17 bajtov. Výsledný uplink rámec má nasledovnú podobu [26]:

Dĺžka	Označenie poľa	Funkcia
19 bitov	UL-Pr	Preamble
13 bitov	FT	Frame Type
2 bity	LI	Length Indicator
1 bit	BF	Bidirectional Flag
1 bit	REP	Repeated Flag
12 bitov	MC	Message Counter
32 bitov	ID	Identifier
0 - 12 bajtov	UL-PAYLOAD	Payload
2 – 5 bajtov	UL-AUTH	Uplink Authentication
16 bitov	UL-CRC	CRC checksum

Tab. 3.3: Formát uplink Sigfox správy

Sigfox umožňuje aj komunikáciu v zostupnom smere (downlink, teda zo základňovej stanice na klienta), kedy má obsah správy dĺžku 8 bajtov (v prípade kratšieho vstupu sú automaticky doplnené nulové bajty) a niektoré z ostatných metadát sú iné. Jej použitie je menej časté – je navrhnutá pre účely ako zmena konfigurácie koncového zariadenia, požiadanie o zaslanie rozšírejších údajov alebo o komunikáciu prostredníctvom vysokorýchlostného spojenia. Zostupná komunikácia môže byť vždy len odpoveďou na vzostupnú správu, nemôže byť iniciovaná samostatne. Komunikačný modul po odvysielaní správy čaká vyhradený čas na odpoveď, potom prechádza do režimu spánku.

Pri rádiovom prenose sú používané modulácie DBPSK (uplink) a GFSK (downlink), pričom každá správa je modulovaná na šírku pásma len 100/600Hz. Frekvenčné spektrum využívané sieťou je 192kHz široké a je možné ho teda pomyselne rozdeliť na 1920 kanálov. Modem si pri vysielaní vyberá náhodne 3 frekvencie v tomto spektre, na ktorých správu odvysiela. Vďaka tejto náhodnosti a faktu, že je vysielanie veľmi úzke je veľmi náročné Sigfox (aj cielene) rušiť.

Znalosť presnej funkcionality systémových polí správy, časovania a ďalších technických detailov nie je pre koncového zákazníka väčšinou podstatná, pretože pri použití už vyvinutých komunikačných modulov a modemov to tvorí úroveň technológie ku ktorej sa užívateľ pri vývoji aplikácie nedostane ani ju nevie konfigurovať. Kompletná špecifikácia protokolu je ale dostupná z oficiálnych zdrojov spoločnosti Sigfox. [27]

Podľa štúdie [28] je pri frekvencii odosielania jednej správy každých 10 minút a pri napájaní 2400mAh batériou predpokladaná životnosť zariadenia približne 2,5 roka. Pri menej častom odosielaní môže teoretická výdrž stúpnuť nad 10 rokov, kedy má zmysel uvažovať už skôr o fyzickej životnosti batérie vzhľadom k jej prirodzenému starnutiu a samovybíjaniu.

3.4.4 Zabezpečenie

Vysielanie v éteri je zabezpečené šifrovaním AES128, pričom všetky správy šifruje firmvér komunikačného modulu automaticky, na základe symetrického MAC kľúča, ktorý má každý Sigfox modem nemenne nastavený od výroby, kombinovaného s PAC kľúčom, ktorý poskytuje distribútor zariadení alebo operátor.

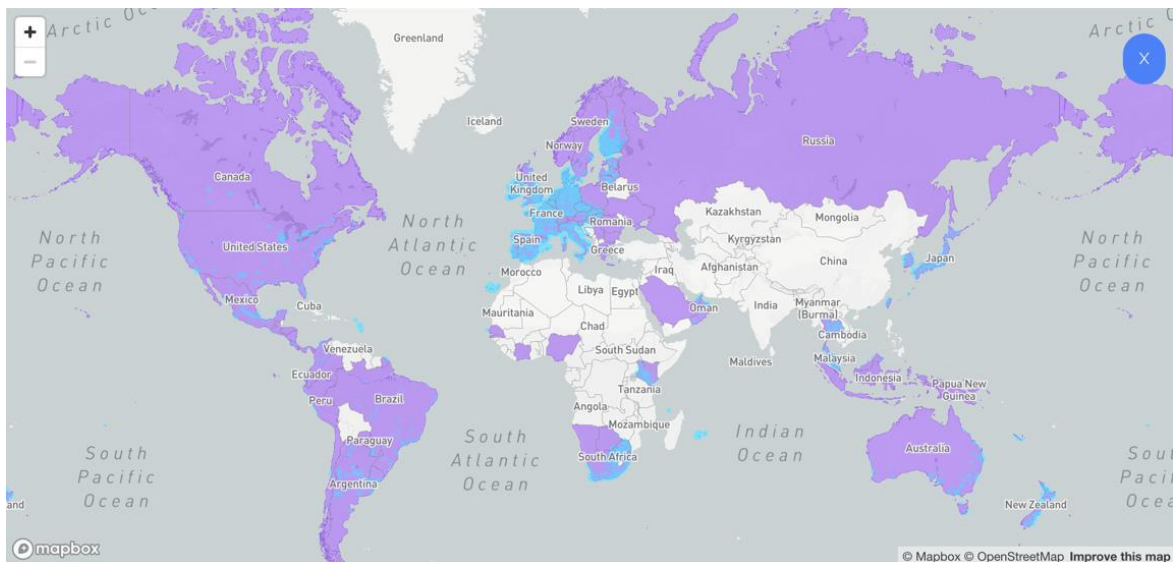
Po úspešnom prijatí vyslanej správy akoukoľvek základňovou stanicou sa overuje CRC kontrolný súčet správy a následne ju stanica odosiela do centrálného Sigfox Cloudu. Pripojenie základňových staníc do internetu je navrhnuté podľa dostupných možností v danej oblasti a je v kompetencii lokálneho operátora, no Sigfox uvádza, že najbežnejšie je použitá DSL technológia v kombinácii s 3G/4G mobilnou sieťou ako zálohou. V prípade dostupnosti je predpokladané uprednostnenie optickej linky, naopak v odľahlých oblastiach sa používa aj satelitné pripojenie. V každom prípade má základňová stanica prostredníctvom virtuálneho tunelu VPN priame pripojenie do serverovej infraštruktúry Sigfox Cloudu. VPN tunel je zabezpečený sadou protokolov IPsec, ktorá šifruje a autentifikuje každý IP paket samostatne, na úrovni samotného IP protokolu. Šifrovaný je celý paket vrátane hlavičky, pričom je pridaná nová hlavička. V rámci OSI modelu sa teda jedná o šifrovanie už na úrovni tretej (sieťovej) vrstvy, ktoré umožňuje bezpečný prenos dát aj nie-dôveryhodnou sieťou. Všetky šifrovacie kľúče sú navyše na strane BTS generované a ukladané pomocou TPM modulu, ktorým musia byť základňové stanice vybavené. TPM modul zároveň zabezpečuje aj integritu operačného systému základňovej stanice.

Sigfox Cloud je centrálny hub pre všetky správy zo všetkých Sigfox zariadení na svete, ktorý prevádzkuje samotná spoločnosť Sigfox. Z dôvodu kapacitných možností a hlavne bezpečnosti nejde o jedno datacentrum ale o sieť viacerých, ktoré spolu komunikujú a navonok sa javia ako jednotná infraštruktúra. Sigfox verejne neuvádza podrobnosti o týchto datacentrách ani ich polohu, nachádzajú sa však na území Francúzska a ide o certifikované datacentrá s adekvátnym zabezpečením na všetkých úrovniach, redundantným zabezpečením konektivity, napájania a na aplikačnej úrovni aj všetkých komponentov, plným load-balancingom a podporou škálovateľnosti v prípade zvýšenej potreby zdrojov. Po prijatí správy overuje Sigfox Cloud sekvenčné číslo (ochrana pred replay útokmi), integritu správy a odosielateľa pomocou MAC kľúča a tiež dešifruje samotný obsah správy.

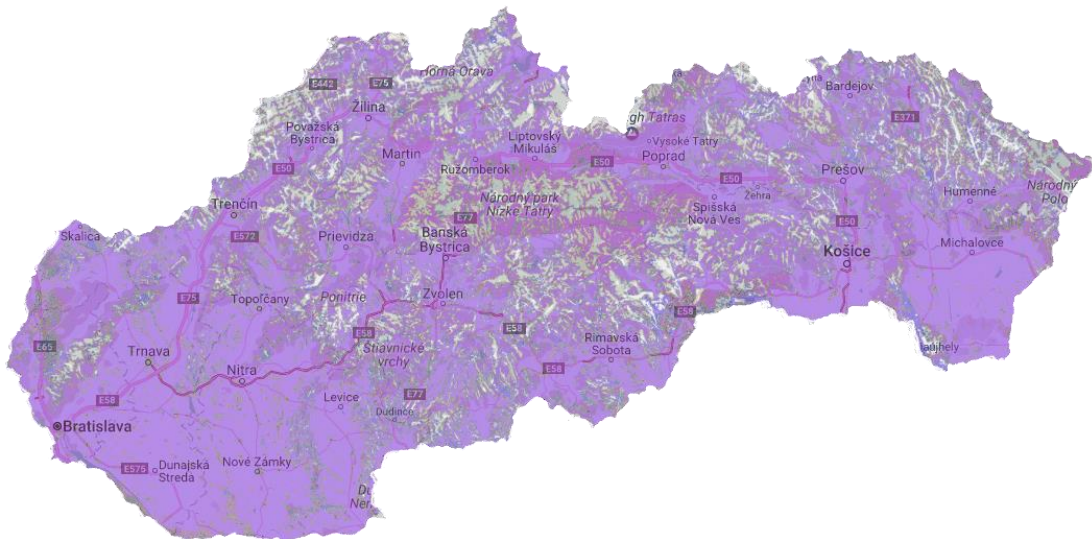
Mechanizmy šifrovania a zabezpečenia sme opísali len zjednodušene, jeho podrobnosti je možné nájsť v oficiálnych zdrojoch spoločnosti Sigfox. Použitá kombinácia zabezpečení však robí technológiu Sigfox veľmi bezpečnou a zneužitie je takmer nemožné.

3.4.5 Dosah a pokrytie

Sigfox aktuálne dosahuje celosvetovo pokrytie 5,8 miliónov km² a 1,3 miliardy obyvateľov v 77 krajinách po celom svete (január 2021) [29,30]. Jeho ďalšie rozširovanie je pritom veľmi rýchle a medziročne stúpa v desiatkach percent. Mapy pokrytia vo svete [31] a na Slovensku [32] môžeme vidieť nižšie, na celosvetovej mape sú modrou farbou aktuálne potvrdené pokryté územia a fialovou farbou krajiny s ktorými spoločnosť rokuje, respektíve už prebieha budovanie sietí, ale overené mapy pokrytia ešte nie sú k dispozícii. Na mape Slovenska je fialovou farbou územie, ktoré je pokryté aspoň jednou základňovou stanicou.



Obr. 3.11: Pokrytie Sigfox siete celosvetovo



Obr. 3.12: Pokrytie Sigfox siete na Slovensku

Ako už bolo spomenuté, v každej krajine prevádzkuje sieť len jeden operátor, ktorý má uzavreté partnerstvo so spoločnosťou Sigfox. Na Slovensku je to spoločnosť SimpleCell Networks Slovakia a.s., ktorá už ale ako operátor vystupuje pod názvom Sigfox Slovakia (v minulosti ako SimpleCell). Tento trend jednotných názvov operátorov, vystupujúcich ako dcérske spoločnosti už nasledovali mnohé krajiny.

Bežný dosah signálu k základňovej stanici je vo voľnej krajine uvádzaný do 50km, v zastavanej oblasti cca 10km, pričom pri dobrej sile signálu, alebo v pokrytí viacerých

staníc súčasne nie je problémom ani komunikácia z interiérov budov alebo dokonca z podzemných priestorov. Často však reálny dosah prevyšuje teoretické očakávania, o čom svedčia aj niekoľkokrát prekonané rekordy. Momentálny rekord z decembra 2016 je zachytenie správy vyslanej GPS trackerom na vrchu hory v Írsku základňovou stanicou vzdialenou 1023 kilometrov, na pobreží Španielska [33]. V prípade potreby dodatočného pokrytia malého územia ponúka Sigfox hardvér takzvanej mikrozakladňovej stanice, ktorá dokáže rozšíriť alebo posilniť pokrytie napríklad v interiéroch budov alebo v priemyselných komplexoch. Udávaný dosah je 1km v exteriéri, prípadne je možné ho ďalej rozšíriť pomocou dostupných opakovačov signálu (repeaterov).



Obr. 3.13: Jednotka mikrozakladňovej stanice Sigfoxu a repeater signálu

Sigfox poskytuje tiež HTTP API rozhranie, pomocou ktorého je možné zistiť predpokladanú úroveň signálu v lokalite určenej GPS súradnicami alebo poštovou adresou. Názov tohto rozhrania je Global Coverage API.

3.4.6 Sigfox Atlas

Podobne ako je v mobilných sieťach možné podľa sily signálu z viacerých základňových staníc možné trianguláciou vypočítať približnú polohu zariadenia, to umožňuje aj Sigfox. Jedná sa o doplnkovú službu pod názvom Sigfox Atlas, kedy Sigfox Cloud v rámci spracovania duplicit prichádzajúcej správy z rôznych základňových staníc

vypočíta na základe úrovne RSSI a známych polôh vysielačov ktoré správu prijali približné umiestnenie, respektíve oblasť, v ktorej sa zariadenie ktoré správu odvysielalo nachádza. Veľkou výhodou tejto možnosti je to, že nevyžaduje žiadny prídavný hardvér ani konfiguráciu navyše a funguje pre všetky aktuálne aj budúce zariadenia v sieti Sigfox. Polohu je možné vypočítať pre každú prijatú správu z hocikakého zariadenia. Presnosť takejto lokalizácie závisí od množstva základňových staníc v danej lokalite, Sigfox uvádza, že pre 80% správ sa presnosť lokalizácie pohybuje v rozsahu 1-10km. Tento údaj samozrejme nemôže konkurovať presnosti lokalizácie pomocou GPS, ale je to lacná a dostupná možnosť približnej lokalizácie, ktorá je veľmi výhodná v aplikáciách kde je veľké množstvo zariadení a presnosť kriticky dôležitá. Dobrým príkladom je sledovanie tovaru pri medzinárodnej preprave. Pre účel lokalizácie môže byť vysielaná správa aj prázdna, a teda pri minimálnej obstarávacej cene zariadení a prakticky nekonečnej výdrži na batériu môže zákazník sledovať polohu každého prepravovaného kontajnera/palety/krabice samostatne. V prípade potreby presnejšej lokalizácie existuje rozšírenie služby s názvom Atlas WiFi, ktoré funguje tak, že zariadenie vyžadujúce spresnenie polohy, ak je vybavené Wi-Fi modulom, zoskenuje okolité Wi-Fi siete a MAC adresy vysielačov (access-pointov) pošle ako obsah správy na Sigfox Cloud. Ten kontaktuje s požiadavkou lokalizačnú službu HERE, ktorá vedie celosvetovú databázu polôh Wi-Fi access-pointov, a tá v prípade, že dané MAC adresy má v databáze, vráti údaje o polohe. Na lokalizáciu týmto spôsobom postačuje aj jediná MAC adresa Wi-Fi vysielača v okolí³, čo je výhodou oproti konkurenčnému poskytovateľovi rovnakej služby – Googlu, ktorý vyžaduje najmenej 2 MAC adresy [34].

3.4.7 Sigfox Cloud

Po prijatí správ vykonáva Sigfox Cloud okrem bezpečnostných funkcií popísaných v predchádzajúcej podkapitole ďalšie úlohy:

- páruje navzájom duplicity tej istej správy prijatej viacerými vysielačmi,
- vykonáva pri každej správe kontrolu, či má vysielajúce zariadenie autorizáciu komunikovať v Sigfox sieti (platný kontrakt, prípadne lokálne obmedzenia),
- ráta metriku pre daného cieľového zákazníka,

³ V prípade poskytnutia len jednej MAC adresy je vrátená poloha s presnosťou len 500m z dôvodu ochrany súkromia. Pri poskytnutí dvoch a viac MAC adries je presnosť najvyššia možná.

- archivuje všetky správy,
- dopĺňa dodatočné údaje (ako napríklad polohu v rámci služby Sigfox Atlas),
- a distribuuje správy ďalej cieľovým adresátom danej aplikácie.

Sigfox Cloud tiež poskytuje administračné rozhranie s názvom Sigfox Backend, do ktorého sa vedia prihlásiť zákazníci (prevádzkovatelia služieb) a zástupcovia operátorov. Po prihlásení je tu možné vykonávať kompletný manažment vlastných zariadení, deliť ich do skupín a sérií, registrovať nové zariadenia, nastavovať privilégia ďalším používateľom (napríklad technická podpora), spravovať kontrakty, sledovať štatistiky, nastavovať upozornenia na rôzne udalosti, pridávať privilégia ďalším používateľom, a hlavne nastavovať tzv. callback metódy (spätné volania) – mechanizmy na preposlanie správy cieľovému adresátovi. V týchto metódach je možné nastaviť kam sa budú prijaté dáta posielat'.

Time	Delay (s)	Header	Data / Decoding	Location	Base station	RSSI (dBm)	SNR (dB)	Freq (MHz)	Rep	Callbacks
2017-04-27 13:51:46	< 1	0000	504F485942202020 ASCI: P0HYB	+	26FE	-129.00	22.36	868.1732	1	+
2017-04-27 13:51:05	1.5	0000	504F485942202020 ASCI: P0HYB	+	26FE	-110.00	40.62	868.1489	3	+
2017-04-27 13:50:48	2	0000	504F485942202020 ASCI: P0HYB	+	26FE	-113.00	38.06	868.1640	1	+
2017-04-27 13:50:27	1.9	0000	504F485942202020 ASCI: P0HYB	+	26FE	-108.00	42.90	868.1047	1	+
2017-04-27 13:50:15	1.9	0000	504F485942202020 ASCI: P0HYB	+	2108	-134.00	16.82	868.1844	1	+
2017-04-27 13:49:57	1.9	0000	504F485942202020 ASCI: P0HYB	+	26FE	-110.00	40.83	868.1840	2	+
2017-04-27 13:49:57	1.9	0000	504F485942202020 ASCI: P0HYB	+	26FE	-110.00	41.20	868.1794	3	+
2017-04-27 13:49:42	< 1	0000	504F485942202020 ASCI: P0HYB	+	26FE	-112.00	38.70	868.1979	3	+
					210B	-133.00	18.42	868.1590	1	+
					268B	-136.00	15.44	868.1587	1	+

Obr. 3.14: Prehľad prijatých správ zo zariadenia v Sigfox Backende

Zariadenia sú rozdelené povinne minimálne podľa “Device Type“, voliteľne môžu byť ďalej rozdelené podľa sérií, a nezávisle priradené do tzv. skupín. Pod Device Type sa rozumie jeden typ zariadenia, poskytovaný napríklad pre jednu službu a toto delenie je kľúčové pre ďalšie nastavenia. Delenia na skupiny a série je užitočné skôr pre administratívnu prehľadnosť. Pre skupiny Device Type sa samostatne nastavujú spomínané callback metódy. Tie sa delia na 2 typy:

- UPLINK, ktoré sú “jednosmerné“, teda správa sa len pošle na cieľovú destináciu
- BIDIR, ktoré môžu (nemusia) byť “obojsmerné“ a vedia sprostredkovať odpoveď zariadeniu – teda downlink komunikáciu

Callback metóda následne pozostáva z nastavenia:

- Typu: DATA (štandardná správa zo zariadenia) / SERVICE (viaceré využitia) / ERROR (chybová udalosť), EVENT (viaceré využitia)
- Dekódovacej gramatiky: voliteľné, pre dekodovanie prenášaných dát
- Kanálu: URL / Batch URL (viacero adries) / E-mail
- Destinácie: URL adresa(y) / mailová adresa
- Správy: telo POST/PUT metódy / obsah emailu
- HTTP hlavičiek: len v prípade URL kanálu

V URL adrese, HTTP hlavičkách aj tele správy je možné používať premenné, vždy dostupné sú ID zariadenia, typ zariadenia, čas, dátum a sekvenčné číslo, a v prípade definovania dekodovacej syntaxe aj všetky vlastné premenné takto vytvorené.

Dekódovacia gramatika je zápis popisujúci ako sú v prenášaných dátach zakódované jednotlivé údaje. Je dôležité si uvedomiť, že surový obsah Sigfox správy je prostá postupnosť bajtov, a teda bez dekodovania je ju možné zobrazit' len ako dlhú hexadecimálnu hodnotu. Je možné použiť jednoduché dekodovanie na text pomocou ASCII tabuľky, súradníc na polohu, alebo napísať vlastnú syntax podľa určenej gramatiky. Základný formát je:

názov_premennej : [index_bajtu] : dátový_typ : [endianita] : [bitový_posun]

Presné pravidlá syntaxe sú na obrázku č. 3.15:

```
format = field_def [" " field_def]*;  
field_def = field_name ":" byte_index ":" type_def ;  
field_name = (alpha | digit | "#" | "_")* ;  
byte_index = [digit]* ;  
type_def = bool_def | char_def | float_def | uint_def ;  
bool_def = "bool:" ("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7") ;  
char_def = "char:" length ("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7") ;  
float_def = "float:" ("32" | "64") [ ":" :little-endian" | ":" :big-endian" ] ("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7") ;  
uint_def = "uint:" ["1" - "64"] [ ":" :little-endian" | ":" :big-endian" ] ("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7") ;  
int_def = "int:" ["1" - "64"] [ ":" :little-endian" | ":" :big-endian" ] ("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7") ;  
length = number* ;  
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

Obr. 3.15: Gramatika dekodovacej syntaxe

Telo správy na odoslanie môže byť formátované ako plain text, JSON objekt, alebo páry kľúč:hodnota.

Url pattern `https://fe6ea208.eu-gb.apigw.appdomain.cloud/ecar-iot-kit-api/v1/sigfox/uplink`

Use HTTP Method `POST`

Send SNI (Server Name Indication) for SSL/TLS connections

Headers

X-Debug-Mode	true	✕
X-AWS-Client-Id	74a9df9b-9b32-49e0-8252-860871a18e01	✕
contentHeader	{time}	✕
X-AWS-Client-Secret	f8c630b5-3acd-456c-991f-daf[REDACTED]	✕
myheader	skuska	✕
header	value	

Content type `application/json`

Body

```
{
  "time": "{time}",
  "deviceID": "{device}",
  "seqNumber": "{seqNumber}",
  "data": "{data}",
  "state": "{customData#state}",
  "current_charge": "{customData#current_charge}",
  "target_charge": "{customData#target_charge}",
  "current": "{customData#current}",
  "elapsed_time": "{customData#elapsed_time}",
  "remain_time": "{customData#remain_time}",
  "range": "{customData#range}",
  "elec_consumption": "{customData#elec_consumption}",
  "indoor_temp": "{customData#indoor_temp}"
}
```

Obr. 3.16: Ukážka callback metódy v Sigfox Backende

Pre najpoužívanejšie cloudové platformy (Amazon AWS, Microsoft Azure, IBM Cloud) sú pripravené šablóny, ktoré vytvorenie callbacku uľahčujú.

Po prijatí každej novej správy z akéhokoľvek zariadenia danej Device Type skupiny sa vytvorená callback metóda spustí a prepošle dáta z tejto správy na nastavenú destináciu, v nastavenom formáte. Ak to daný kanál umožňuje (napr. URL), je k správe archivovaná aj návratová hodnota a vrátené údaje z príslušného volania.

Okrem mechanizmu callbackov poskytuje Sigfox Cloud aj REST API, pomocou ktorého je možné manažovať a registrovať zdroje z externej služby, získať obdržané správy zo želaného zariadenia v ľubovoľnom čase atď.

Sigfox je v neustálom vývoji, pravidelne vychádzajú nové aktualizácie častí siete a Sigfox Cloudu, ktoré ďalej optimalizujú ich vlastnosti a prinášajú nové funkcionality. To mu umožňuje oproti konkurencii možnosť flexibilnejšie reagovať na potreby trhu, zákazníkov a rôzne zmeny v technologickom svete. Cena kontraktov sa odvíja od množstva zariadení, povoleného denného počtu správ a aktivácie prídavných funkcií (Monarch, geolokácia, ...), no pohybuje sa v ráde jednotiek eur ročne za jedno zariadenie. Tiež sú dostupné účtovacie modely Pay-as-you-grow, fixné balíky a vývojárske prístupy pre malé množstvo zariadení.

3.5 Porovnanie vhodnosti pre EV a navrhnuté riešenie

Bezdrôtové pripojenie elektronických zariadení k internetu je výrazne sa rozrastajúcim trhom, na ktorom dnes existuje veľa rôznych technológií od rôznych výrobcov a aliancií, ktoré sa od seba ale značne odlišujú. Konceptia IoT prináša pre rôzne typy zariadení rozdielne požiadavky a zároveň je tento trend relatívne nový, preto je prirodzené, že v súčasnosti tieto technológie vznikajú ale aj zanikajú a trh sa formuje. Niektoré aplikácie vyžadujú vysokú prenosovú rýchlosť či nízku odozvu, pri iných zas postačuje minimálna rýchlosť, no kľúčová je spotreba energie. Vlastnosťami, ktorých dôležitosť sa medzi rôznymi aplikáciami líši je tiež spoľahlivosť doručovania respektíve odolnosť voči rušeniu, (ne)možnosť prepájania medzi viacerými sieťami/bunkami, roaming, podpora prenosu hlasu, lokalizácie, spätného prenosu (downlink), (ne)vhodnosť pre pohybujúce sa objekty, podpora lokalizácie, či možnosť prispôbovať prenosovú rýchlosť podľa aktuálnej potreby. Nie je možné, aby jedna technológia splnila všetky tieto rôzne, často nezlučiteľné požiadavky. Často si rôzne technológie ani nekonkurujú, cieľia na iné aplikácie, a v prípade potreby sa navzájom dopĺňajú. Všeobecne sa dá povedať, že pre lokálne IoT aplikácie sú teoreticky najvhodnejšie WLAN technológie zo skupiny štandardov IEEE 802.11. Pre aplikácie IoT fungujúce na väčšom území (napríklad riešenia smart city) sú vhodnejšie komunikačné technológie typu LPWAN, ktoré združujú vlastnosti ako malé množstvo prenášaných dát a malá prenosová kapacita, no vysoký dosah, nízka energetická náročnosť a jednoduchý hardvér. Pre globálne dostupné IoT aplikácie, alebo také, ktoré potrebujú prenášať väčšie množstvá dát sú zas k dispozícii mobilné siete založené na technológiách GSM.



Obr. 3.17: Logá štyroch najväčších technológií z oblasti LPWAN

V prípade moderných automobilov sú požiadavky na ich konektivitu mimoriadne komplexné. Jednotlivé funkcionality majú rozdielne požiadavky:

- V2V a V2X – minimálna odozva, point-to-point, malé množstvo dát, postačuje krátky dosah
- Infotainment – vysoká rýchlosť, veľké množstvo dát, neustála dostupnosť nie je kritická
- Vzdialené pripojenie a sledovanie – dostupnosť a spoľahlivosť, malé množstvo dát, stačí nízka až stredná rýchlosť, rýchlosť odozvy nie je kritická

Vhodným (a jediným) riešením je preto skombinovať použitie viacerých prenosových technológií súčasne. Keďže táto práca sa zameriava hlavne na prenos informácií o nabíjaní elektromobilu, čo spadá do tretieho bodu, analyzovali sme najmä LPWAN technológie. Požiadavkou bola dostupnosť sietí na území Slovenska a dostupnosť komunikačných modulov. Túto podmienku spĺňajú všetky 4 zo súčasne najrozšírenejších LPWAN technológií, no vylúčili sme NB-IoT, pretože kvôli obmedzeným možnostiam prepájania medzi bunkami je nevhodné pre mobilné aplikácie. Výber medzi zvyšnými tromi možnosťami (LoRa, Sigfox a LTE-M) bol veľmi náročný. Všetky 3 z nich ponúkajú isté výhody aj nevýhody oproti ostatným a pri vývoji konkrétneho komerčného produktu by bolo potrebné ich opäť dôrazne prehodnotiť. Náš finálny výber bola technológia Sigfox z nasledujúcich dôvodov:

1. Globálna dostupnosť vo väčšine sveta
2. Automatický a bezplatný roaming vo všetkých krajinách
3. Výborné reálne pokrytie a dosah od základňovej stanice
4. Možnosť automatického prepínania medzi rôznymi ISM pásmami
5. Prevádzkovanie nadnárodnou komerčnou spoločnosťou

Treba ale spomenúť tiež nevýhody – limitovaný počet správ za deň, ich limitovaná veľkosť a veľmi nízka prenosová rýchlosť.

Tie by odpadli v prípade technológie LoRa, nahradili by ich však iné – nemožnosť roamingu v prípade použitia komerčných sietí respektíve otázna spoľahlivosť v prípade použitia komunitnej siete The Things Network a tiež v budúcnosti možné problémy pri veľkom počte zariadení na menšom území, napríklad v husto osídlených oblastiach.

Tieto nevýhody spoločne aj s tými zo Sigfoxu by odstránilo použitie licencovaného LTE-M, ktoré však môže byť výrazne drahšie, bolo by potrebné používať SIM karty a dostupnosť v zahraničných krajinách a roaming sú otáznе. Taktiež na Slovensku bola dostupnosť a pokrytie v čase začiatku písania tejto práce ešte len v štádiu testovacej prevádzky.

Pri použití Sigfoxu (a použití Sigfox Monarch kompatibilného modemu) sa stáva navrhnuté riešenie automaticky funkčným vo všetkých krajinách kde je (bude) Sigfox dostupný, čo majiteľovi automobilu umožňuje používať aplikáciu aj pri cestovaní kamkoľvek do zahraničia, bez potreby čokoľvek nastavovať. Dôležitou výhodou je tiež vysoký dosah od základňových staníc, čo môže umožniť konektivitu aj v odľahlejších oblastiach, napríklad nabíjajúcich staniach mimo miest alebo v podzemných garážach. Dostupnosť LoRa staníc sa sústreďuje v mestách, LTE siete (na ktorých je závislé LTE-M) tiež nemusia v niektorých krajinách dosahovať tak vysoké percento územia. Predpokladané pokrytie je tiež možné overovať pomocou dostupného API, vďaka čomu je možné majiteľovi vozidla zobraziť upozornenie v prípade, že sa aktuálne nachádza na území kde pokrytie pravdepodobne nie je. Treťou podstatnou výhodou je, že Sigfox je globálne prevádzkovaný jednou spoločnosťou (a siete jej partnermi), čo pre medzinárodnú aplikáciu znamená garanciu spoľahlivosti a dostupnosť podpory. Nezanedbateľná je tiež jednoduchosť implementácie, kedy po prvotnom nastavení komunikačného modulu/modemu nie je potrebná žiadna ďalšia správa komunikačnej časti aplikácie, ani na strane koncového zariadenia, ani siete. Pomocou doplnkovej služby Sigfox Atlas by tiež bolo implementovať funkciu núdzovej lokalizácie vozidla, napríklad v prípade krádeže a rušenia GPS signálu.

Spomínané nevýhody nemusia byť až také závažné – maximálny payload jednej správy je síce len 12 bajtov, no pri dobre navrhnutej kompresii je možné aj do tohto limitu vmestiť relatívne veľké množstvo informácií. Pre náš systém sme navrhli 2 typy správ (regulárna a rozšírená) pomocou ktorých vieme prenášať až 13 rôznych parametrov, vrátane GPS súradníc. Formát týchto správ už bol popísaný v kapitole 2.2.

Druhou, vážnejšou nevýhodou je obmedzenie počtu správ na 140 za deň (vo vzostupnom smere). Pri pravidelnom nepretržitom vysielaní by teda interval mohol byť najviac 6 správ za hodinu – aktualizácia každých 10 minút. V našej aplikácii ale nepotrebujeme prenášať dáta celý deň, iba keď je vozidlo pripojené na nabíjanie a nie je

dostupná iná konektivita (vysvetlené ďalej). Nami doporučené nastavenie je rezervovať 20 správ denne pre rozšírené správy a posielať aktualizáciu (regulárnu správu) pri nabíjaní každých 5 minút, čo činí 12 správ za hodinu. To umožní dodržať limit pri nabíjaní najviac 10 hodín denne (120 správ za deň), čo považujeme za dostačujúce. Pri intervale vysielania každých 2,5 minúty by limit vystačil na 5 hodín nabíjania za deň, na verejných nabíjačkách. Rozšírené správy navrhuje naša schéma vysielat' jednorazovo po pripojení nabíjania, respektíve pri mimoriadnej udalosti (upozornenie na prekročenie bezpečnej teploty a prítomnosť osoby/zvieraťa vo vozidle, spustenie alarmu, označenie vozidla ako ukradnutého, atď.). Pri takomto nastavení je nepravdepodobné, aby došlo k prekročeniu požadovaných limitov. V prípade potreby by bolo možné dosiahnuť aj vyššie množstvá správ obmedzením počtu vysielaní jednej správy (štandardné nastavenie je 3x) a tým skrátením dĺžky vysielania⁴.

Keďže je však predpoklad, že v značnom počte prípadov nabíja majiteľ svoj elektromobil doma, je náš systém doplnený aj o možnosť konektivity prostredníctvom lokálneho pripojenia Wi-Fi (802.11n). V prípade že riadiaca jednotka rozpozná, že je dostupná známa sieť Wi-Fi, automaticky využíva na všetku komunikáciu toto pripojenie a je možné posielať aktualizácie v reálnom čase. Za známu sieť sa pritom nemusí považovať len domáca Wi-Fi, ale môže ísť aj sieť na partnerských nabíjajúcich staniciach, s ktorými by mal teoretický prevádzkovateľ riešenia dohodu. Automobil by sa na týchto staniciach vedel pripojiť na danú sieť automaticky, bez potrebného zásahu užívateľa.

Na prístup do siete Sigox sme použili komunikačný modul "LPWAN SigFox node" od českého výrobcu LPWAN/cz, ktorý je založený na modeme SFM10R1 a malú anténu so ziskom 3dBi. Ide o modem podporujúci regióny RC1, RC6 a RC7 (Európa, Rusko, India, Afrika), v ktorých sa používa ISM pásmo 868MHz. Cena modulu sa pri maloobchode pohybuje okolo 15eur. Napriek malej anténe sme pri rýchlych testoch dosiahli v krajských mestách vždy takmer plnú kvalitu signálu. Pri použití väčšej antény so ziskom 8dBi sa nám prenos podaril aj z doliny na vidieku, z miesta mierne za hranicou oficiálnej mapy pokrytia, vzdialeného od základňovej stanice v kopcovitej oblasti približne 25km. Vo vozidle sa počíta s možnosťou použitia aj väčších antén, kedy by

⁴ Pravdepodobne by tiež bola potrebná dohoda s lokálnym operátorom

dosah mohol byť ešte väčší ako sa bežne predpokladá. Použitá rádiová konfigurácia v našom simulátore je štandardná:

- počet opakovaní vysielania – 3x
- vysielacia a prijímacia frekvencia – automaticky
- vysielací výkon – 14dBm (max)

4 CLOUD BACKEND

Cloud je veľmi dôležitou súčasťou väčšiny aplikácií zo sveta Internetu vecí. Dáta, ktoré IoT senzory a zariadenia produkujú je potrebné spracovať a na základe ich vyhodnocovania vykonávať rozhodnutia o akciách ktoré v systéme vykonávajú aktuátory, alebo je ich potrebné ukladať pre dlhodobejšie vyhodnocovanie. Túto logiku a úložné možnosti poskytuje backend aplikácie, ktorý môže bežať na lokálnej serverovej infraštruktúre prevádzkovateľa aplikácie (napríklad pokiaľ ide o aplikáciu na úrovni firmy), alebo na distribuovanej infraštruktúre v internete. V prvom prípade potom hovoríme o On-premises riešeniach, v druhom prípade o cloud-computingu.

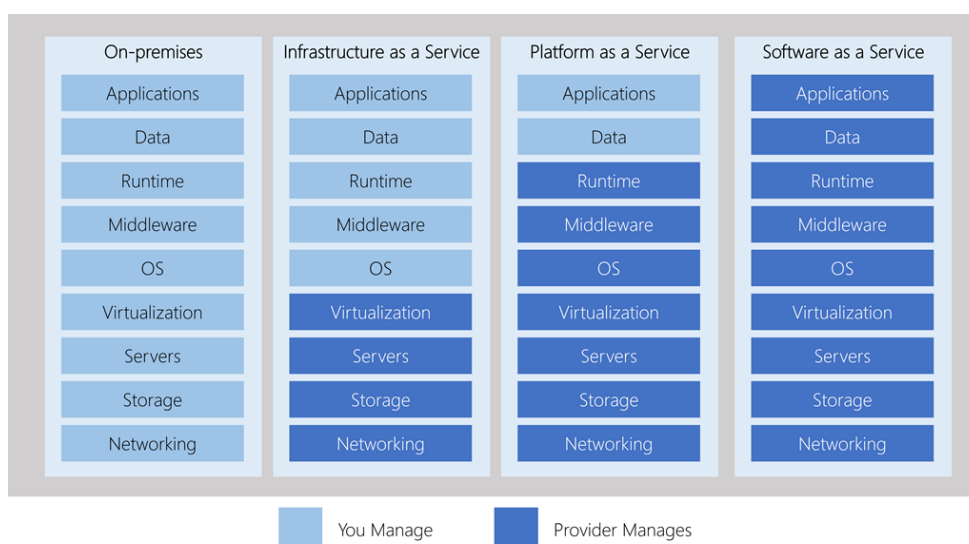
Poskytovateľ cloudových služieb prevádzkuje celú infraštruktúru – stará sa o jej údržbu a dostupnosť, čo zahŕňa obmenu hardvéru a aktuálnosť softvéru, napájanie, konektivitu a fyzické aj informačné zabezpečenie, a následne túto infraštruktúru prenajíma svojim zákazníkom. Pod infraštruktúrou sa môže rozumieť všetko od hardvérových prostriedkov (servery, úložiská, sieťové zariadenia), cez softvérové platformy (logická správa hardvéru, operačné systémy, databázové servery), až po konkrétny softvér (CMS systém, poštový server, aplikácie). Podľa toho, čo je predmetom prenájmu, sa rozlišujú 3 základné distribučné modely [40]:

- *IaaS (Infrastructure-as-a-Service)* – Infraštruktúra ako služba, je model kedy je prenajímaná infraštruktúra v podobe (zdanlivo) samotného hardvéru – servery, sieťové prvky, dátové úložiská, atď. V skutočnosti sa ale používa virtualizácia, teda používateľ nemá prístup k fyzickým zariadeniam poskytovateľa, ale k virtuálnej infraštruktúre. Po technickej stránke to nepredstavuje žiadny rozdiel, avšak umožňuje to jednoduchú správu a škálovateľnosť – od malého výkonu pre vývoj/test aplikácie, cez spustenie do reálneho nasadenia, po pokrytie požiadaviek desiatok tisíc prístupujúcich klientov.
- *PaaS (Platform-as-a-Service)* – Platforma ako služba, je model kedy je poskytovaná prednastavená platforma, na ktorej môže zákazník vyvíjať alebo hostovať už vytvorené aplikácie. Ide o zatiaľ najmladší koncept cloudového modelu, ktorého definícia nie je úplne zrejmá. Platformu môže predstavovať prostredie v ktorom bežia aplikácie (Docker, Google App Engine, webhosting), sadu nástrojov na ich vývoj (Kubernetes, GitHub, ...), ale aj celé portfólio služieb,

z ktorých má zákazník možnosť využívať tie, ktoré sú potrebné pre jeho aplikáciu (Amazon Web Services, Microsoft Azure, Google Cloud). PaaS sa v realite často prekrýva s inými modelmi.

- *SaaS (Software-as-a-Service)* – Softvér ako služba, je model kedy poskytovateľ prenajíma už vytvorené aplikácie. Vlastníkom licencií k softvéru teda nie je používateľ, platí len za ich používanie. SaaS sa väčšinou používa vo firemnom prostredí a ako príklad môže slúžiť balík Google Docs alebo Microsoft Office 365, cloudové úložiská pre širokú verejnosť ako Dropbox a OneDrive, a pri širšej definícii aj všetky webové aplikácie.

V každom z týchto modelov je medzi poskytovateľom a užívateľom iné rozdelenie zodpovednosti za správu jednotlivých častí systému [36]:



Obr. 4.1: Porovnanie distribučných modelov cloudu a On-premises riešenia

Okrem spomenutých troch modelov existuje aj nespočetné množstvo ďalších, ktoré však väčšinou nezapadajú do typového delenia. Pri označovaní sa používa konvencia XaaS, kedy čokoľvek môže byť poskytované ako služba (X) – hardvér, sieť, databáza, umelá inteligencia, API, firewall, bezpečnosť, monitoring, hra, a tak ďalej. Zákazníci platia za využívanie služieb mesačné alebo ročné poplatky, ktoré sa odvíjajú od používaných komponentov a využitej kapacity – veľkosti úložného priestoru, prenajatého výkonu, atď.

Druhý spôsob delenia cloud-computingu je podľa toho, na akého cieľového zákazníka poskytovateľ mieri. Podľa toho rozlišujeme modely nasadenia [40]:

- *Verejný cloud* – je k dispozícii pre širokú verejnosť a jeho služby môže využívať každý
- *Privátny cloud* – je prevádzkovaný pre konkrétneho zákazníka a jeho súkromné účely
- *Hybridný cloud* – ide o kombináciu, kedy je súkromná časť, alebo On-premises infraštruktúra napojená na služby verejného cloudu
- *Komunitný cloud* – model, kedy je infraštruktúra zdieľaná medzi viacerými organizáciami, ktoré väčšinou spája rovnaký záujem

Hoci je cloud-computing v súčasnosti jeden z najväčších trendov vo svete IT a je kľúčovým prvkom v moderných aplikáciách, v skutočnosti to nie je nový koncept. Princípy cloud-computingu sú známe a využívané už od rozmachu samotného internetu – dobrým príkladom je poštový klient vo webovom prehliadači, ktorý je vlastne službou typu SaaS. V súčasnej dobe však (aj kvôli IoT) počet aplikácií rastie obrovským tempom a mnoho ľudí ani netuší, že už dlhú dobu aj súkromne využívajú cloud v mnohých službách. Najväčšou výhodou, ktorú cloud-computing prináša pre firmy je, že odpadá nutnosť vlastniť a starať sa o IT infraštruktúru dátových centier. Pre začínajúce a malé firmy to predstavuje značné zníženie vstupných nákladov pred spustením aplikácie, pričom je možné požiadavky neskôr plynulo škálovať. Veľké firmy zas môžu ťažiť z vysokej bezpečnosti ktorú cloudy prirodzene poskytujú, keďže ich prevádzkovatelia sú často lídrami v oblasti a zabezpečenie a dostupnosť služieb je pre nich maximálna prioritou. Servery sú zabezpečené mnohými úrovňami, od kontroly fyzického prístupu, cez sieťovú ochranu, prítomnosť všetkých bezpečnostných záplat, redundanciu napájania hardvéru a konektivity, až po redundanciu všetkých softvérových komponentov. Toto zálohovanie môže prebiehať na lokálnej úrovni v rámci jedného datacentra, alebo na geografickej úrovni, kedy v prípade výpadku celého datacentra môže okamžite prebrať jeho úlohy iné, nachádzajúce sa v inom štáte. Garantovaná dostupnosť služieb pre koncových zákazníkov sa dohaduje v SLA zmluvách a udáva v percentách času ročne. Zaužívaná je konvencia tzv. “počtu deviatok“ – napríklad úroveň 99,9% sa označuje ako tri deviatky a garantuje čas kedy služba nie je dostupná (downtime) maximálne 8,77 hodiny za rok. Pri vyšších úrovniach ako napríklad 99,999% – 5 deviatok, môže byť maximálna nedostupnosť 5,26 minúty za rok, v prepočte necelú jednu sekundu denne.

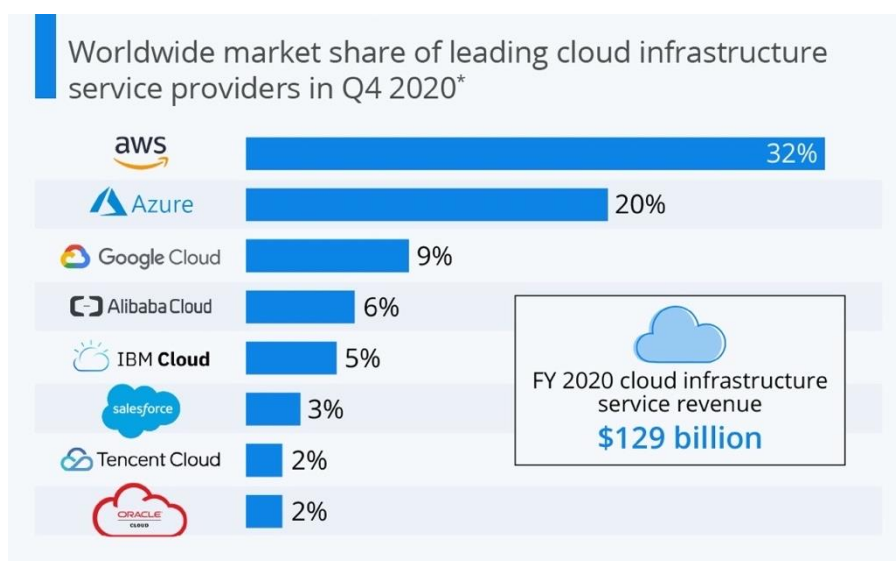
Pre niektoré kritické aplikácie napríklad z vládnej alebo armádnej sféry môže byť problematická legislatívna otázka zverenia správy dát organizáciám tretej strany. Niektorí cloudoví poskytovatelia však vlastní prestížne bezpečnostné certifikáty, vďaka ktorým je možné využitie ich služieb aj v týchto oblastiach.

Najväčšími poskytovateľmi cloudových služieb sú v súčasnosti vedúce technologické spoločnosti v IT svete:

- Amazon – so svojim riešením Amazon Web Services (AWS),
- Microsoft – platforma Azure,
- Google – prevádzkuje Google Cloud Platform a podmnožinu Google Firebase,
- IBM – prevádzkuje IBM Cloud,
- Oracle, Alibaba, SAP, a ďalší...

Všetky riešenia z vyššie menovaných ponúkajú všetky typy cloudových modelov – privátne, hybridné aj verejné, aj základné distribučné modely – IaaS, PaaS a SaaS, pričom sú vzájomne rôzne kombinovateľné podľa potrieb zákazníka.

Množstvo ďalších menších poskytovateľov sa špecializuje na konkrétnejšie oblasti. V prípade zamerania na IoT poskytujú riešenia pre lepšiu konektivitu so zariadeniami, nástroje pre správu a vizualizáciu nazbieraných dát, alebo aj priamo vývojové sety, tiež je dôležitá integrácia na ďalšie cloud platformy a externé služby. Môžeme spomenúť riešenia ako Bosch IoT Suite, SAP HANA, Samsung ARTIK, či Predix (špecializovaný na IIoT).



Obr. 4.2: Rozdelenie trhu cloudových prevádzkovateľov (2020) [37]

Výber hlavnej cloudovej platformy pre použitie v našom riešení nebol jednoduchý. Všetci prevádzkovatelia sa predbiehajú v tvrdeniach, že dokážu pripojiť všetko (IoT zariadenia), ponúknuť akékoľvek služby, bezpečne uložiť dáta a poskytnúť vysokú dostupnosť za skvelé ceny. Ako sme sa ale presvedčili, nie vždy je všetko z toho pravda. Zhrnutie našich poznatkov je nasledovné:

Najprv sme sa kvôli menším predchádzajúcim skúsenostiam a intuitívnemu, prehľadnému grafickému prostrediu administratívneho rozhrania rozhodli použiť IBM Cloud. Po nejakom čase sme si ale zistili, že veľa funkcií nie je úplne doladených a niektoré operácie sa vykonávajú komplikovane. Veľkým negatívom, ktoré nás značne brzdilo pri vývoji bola roztrieštená dokumentácia, ktorá je rozdelená do asi troch rôznych webových knižníc. Niekoľkokrát sa nám stalo, že informácie v nich boli dokonca neaktuálne, alebo nekorešpondovali s aktuálne nasadenou verziou grafického rozhrania. Počas približne dvoch mesiacov sa nám viackrát stalo, že nejaká služba mala výpadky a nefungovala. Finálnym problémom bolo ale zistenie, že komponent IoT brány (Watson IoT Platform), ktorú sme chceli použiť na prijímanie údajov zo zariadení nedokáže v bezplatnej verzii (Lite Plan) komunikovať v reálnom čase s inými komponentami platformy a cena štandardného plánu sa pohybuje okolo 2000 dolárov, pričom v minulosti to možné bolo a na túto zmenu politiky sa v dokumentácii nenachádzalo žiadne upozornenie.

Následne sme (v rámci inej práce) nadobudli isté skúsenosti s platformou Microsoft Azure. Túto platformu hodnotíme ako oveľa vyspelejšiu, no pocit z používania je taký, že sa zameriava skôr na veľké, korporátne riešenia.

Rozhodli sme sa preto vyskúšať aj Amazon Web Services, ktorý je líder na trhu cloudových platforiem a je na trhu najdlhšie. Hoci používa mierne odlišnú koncepciu fungovania (nie je dostupný centrálny dashboard, komponenty sú navonok viac samostatne stojace), hodnotíme ho ako najviac vyspelý. Jednotlivé komponenty majú množstvo možností nastavenia fungovania a monitorovania, ktoré by v prípade reálneho nasadenia umožnili pokročilú optimalizáciu a pohodlné spravovanie. K AWS je dostupná kvalitná dokumentácia a v prípade potreby sa dá spoľahnúť aj na komunitu vývojárov a archívy riešených problémov. Taktiež je dostupný bezplatný balík (Free Tier – Always Free) všetkých potrebných služieb so štedro nastavenými kvótami.

Cloudy, ktoré naše riešenie finálne používa rozoberajú nasledujúce podkapitoly.

4.1 Sigfox Cloud

Do tejto časti už presahovala podkapitola 3.4.7, keďže Sigfox Cloud je vlastne prvým cloudom, ktorého služby naše riešenie používa.

Dáta prijaté do Sigfox Cloudu sú ukladané do proprietárnej databázy, ku ktorej je v prípade potreby prístup možný pomocou administračného rozhrania alebo cez REST API. Náš systém ale používa mechanizmus callback metód. Vytvorená metóda hneď po prijatí správy vykoná dekodovanie obsahu pomocou stanovenej gramatiky (ak je to možné), skonštruuje JSON dokument s týmito údajmi, pridá niekoľko ďalších atribútov (časová značka, id zariadenia, raw dáta), pripojí autorizačné HTTP hlavičky a odošle správu HTTP metódou POST na endpoint REST API na AWS (vysvetlené ďalej).

```
state:0:uint:3:big-endian:7 current_charge:0:uint:7:big-endian:4 target_charge:1:uint:7:big-endian:5
current:2:uint:10:big-endian:6 elapsed_time:3:uint:13:big-endian:4 remain_time:5:uint:13:big-
endian:7 range:6:uint:11:big-endian:2 elec_consumption:8:uint:16:big-endian:7
indoor_temp:10:uint:16:big-endian:7
```

Obr. 4.3: Dekódovacia gramatika regulárnej správy z vozidla

4.2 Amazon Web Services

Ako druhý a hlavný cloud, ktorý náš systém používa na uskladňovanie a spracovanie údajov sme teda použili Amazon Web Services (ďalej AWS). AWS vznikol v roku 2006, keď spoločnosti Amazon nevyhovovalo, že využíva iba 10% dostupnej kapacity svojich datacenter a zvyšok stojí pripravený len pre prípady nárazového využitia (špičiek). Rozhodol sa teda oživiť už v tej dobe vyše 2 desaťročia starú myšlienku zdieľania výpočtového výkonu dátových centier, a vytvoril prvú platformu komerčného cloud-computingu. O rok neskôr predstavil svoje riešenie Google a o ďalší rok IBM, no AWS je od svojho vzniku na trhu cloudových poskytovateľov jednoznačnou dominantou. Používateľom poskytuje veľké množstvo rôznych typov služieb, ktoré sú implementované ako samostatné komponenty, ktoré vedia navzájom komunikovať. Vzniká tak modulárny systém, kde sú od seba jednotlivé časti prevádzkovo nezávislé, účtovanie za ich používanie je vyhodnocované samostatne, je možné ich škálovať, vytvárať redundantné riešenia a integrovať službu s rôznymi inými, aj externými cieľmi. Používateľské účty môžu byť súčasťou organizácií, alebo môžu fungovať samostatne.

V prípade organizácií sú účty zaradzované do stromových hierarchií s rôznymi oprávneniami k jednotlivým zdrojom, ktorých úrovne môžu reflektovať jednotky z reálneho sveta (spoločnosť, trh, pobočka, oddelenie, tím, osoba, ...).

Služby, ktoré potrebujeme od cloudu v našom systéme my, sú v princípe 4: API rozhranie, výpočtové prostredie, databáza a rozhranie pre komunikáciu pomocou technológie push správ.

4.2.1 Výpočtové prostredie

V modernom cloud-computingu sa sformovali 3 prístupy, ako poskytovať výpočtovú kapacitu pre beh samotnej logiky nejakej aplikácie. Tieto prístupy v podstate reflektujú princípy troch spomínaných distribučných modelov cloud-computingu (IaaS, PaaS, SaaS) a líšia sa tým, aká zodpovednosť pripadá prevádzkovateľovi aplikácie a aká prevádzkovateľovi cloudu.

4.2.1.1 Virtuálne stroje

Najstarší, "klasický" prístup predstavuje prenájom virtuálnych strojov a reflektuje princíp IaaS. Prevádzkovateľ cloudu poskytne zákazníkovi prázdny alebo nakonfigurovaný virtuálny stroj, ktorého celý nastavený výkon je zákazníkovi neustále k dispozícii a je čisto na ňom, aké služby na ňom bude prevádzkovať. Zákazník je taktiež sám zodpovedný za správu operačného systému, aktualizácie softvéru, inštaláciu aplikácií, bezpečnosť na aplikačných vrstvách, atď. Nevýhodou tohto riešenia je nízka efektivita – väčšinou sa nevyužíva plný pridelený výkon stroja, no za alokované prostriedky je potrebné platiť celý čas. Ťažko sa tiež dosahuje izolovanosť jednotlivých procesov alebo služieb – v prípade potreby je možné vytvoriť viac virtuálnych strojov, ale to je ekonomicky a administračne nevýhodné. Niekedy potrebnou vlastnosťou ale je, že zákazník má plnú kontrolu nad celým prostredím v ktorom jeho aplikácia beží a neobmedzené možnosti využitia.

4.2.1.2 Kontajnery

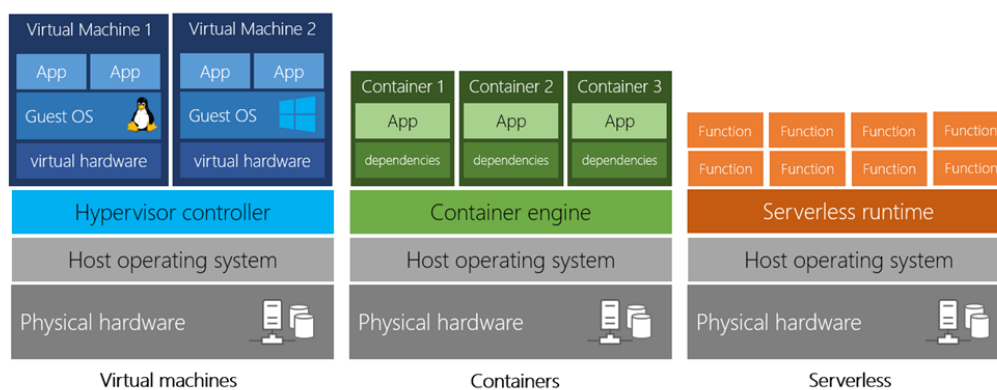
Kontajnerizácia predstavuje moderný spôsob distribúcie aplikácií vo vývojovom a behovom prostredí, ktorá prináša veľa výhod pri modernom agilnom vývoji. Kontajner predstavuje zapuzdrenie aplikácie, podporného softvéru a všetkých jej závislostí tak, aby

sa dala jednoducho prenášať ako celok, na rôzne platformy, a bez nutnosti inštalácie na cieľovom systéme. Aplikácia v kontajneri pracuje samostatne a izolovane, nemá žiadnu závislosť na systéme, v ktorom je kontajner spustený. Platí pravidlo, že jeden kontajner by mal obsahovať iba jednu službu. Ide teda o typ virtualizácie, kedy ale nie je virtualizovaný celý stroj alebo operačný systém, iba štandardizované behové prostredie pre spustenie kontajnera. Takýto prístup umožňuje vďaka prenositeľnosti vyššiu efektivitu vývoja a vďaka izolovanosti a samostatnosti oveľa vyššiu efektivitu využívania zdrojov. V kontexte cloud-computingu predstavuje princíp poskytovania prostredia pre beh kontajnerov distribučný model PaaS. Užívateľovi odpadá nutnosť starať sa o hosťateľský operačný systém na ktorom aplikácia beží, dodá len kontajnery ktoré sú v poskytnutom prostredí spustené, a zaplatí za výpočtové kapacity, ktoré alokujú. Zostáva ale nevýhoda potreby platiť alokované zdroje bez ohľadu na to či je aplikácia aktuálne využívaná alebo nie.

4.2.1.3 *Serverless*

Ide o najvyšší stupeň abstrakcie, kedy je aplikácia dekomponovaná na množstvo drobných častí – bezstavovo navrhnutých funkcií, ktoré navzájom komunikujú a dokopy poskytujú celkové riešenie. Tieto funkcie bežia v prostredí, ktorého správu má plne na starosti prevádzkovateľ cloudu. Ten sa teda stará o všetky hardvérové a softvérové zdroje, užívateľ dodá len samotný kód funkcií v podporovanom programovacom jazyku. Týmto prístupom sa dosiahne maximálna efektívnosť procesu – funkcie nealokujú žiadne prostriedky v čase, keď nebežia. Pri vykonávaní nejakej operácie sú zavolané a spustené len tie funkcie, ktoré sú k vykonaniu danej úlohy potrebné, a sú účtované spotrebované kapacity len po dobu, kedy sa funkcie vykonávali. Súčasne je ale možná neobmedzená škálovateľnosť – hosťateľská platforma dokáže spustiť toľko inštancií funkcií koľko je potrebné, čo umožňuje pokryť ľubovoľne veľké špičky v záťaži bez potreby mať vopred alokované nejaké prostriedky. Zároveň je možná veľmi jednoduchá aktualizácia častí aplikácie, kedy stačí poskytnúť nový zdrojový kód danej funkcie a pri jej najbližšom spustení bude už použitý ten. Nevýhodami tohto riešenia sú potreba navrhnuť aplikáciu od začiatku vývoja pre tento model fungovania, ťažšie testovanie a prenositeľnosť, možná dlhšia doba inicializácie funkcií po zavolaní a zvyčajne obmedzenia na čas vykonávania jednej funkcie. Úlohy pre jednotlivé funkcie by teda mali byť jednoduché a funkcie by sa nemali volať navzájom, čo sťažuje implementáciu zložitejších programov. Výhodami sú

ale nulová administrácia a náklady vyplývajúce z počtu spustení, respektíve nulové náklady pri nečinnosti. Pojem serverless odkazuje na myšlienku, že pre beh aplikácie (sa) zákazník nepotrebuje (starat' o) žiadny server, dodá len funkciu. Preto sa používa aj označenie FaaS (Function-as-a-service).



Obr. 4.4: Rôzne prístupy poskytnutia výpočtových kapacít v cloud-computingu

Nie je možné určiť všeobecne lepší alebo horší z týchto prístupov – ich vhodnosť závisí od konkrétneho prípadu použitia, nakoľko každý z nich má nejaké výhody a nevýhody. V našom systéme sme sa rozhodli použiť prístup Serverless, ktorý na AWS zabezpečuje platforma tzv. Lambda funkcií. Ako behové prostredie sme vybrali Node.js, nakoľko kód funkcií sme poskytli v jazyku JavaScript.

Dokopy používame 5 Lambda funkcií, ktorých úlohy prislúchajú metódam dostupným na REST API (vysvetlené v podkapitole 4.2.3):

- príjem nového stavu z vozidla, uloženie do databázy a preposlanie klientom
- poskytnutie posledného stavu pre želané vozidlo
- poskytnutie rozsahu stavov pre želané vozidlo zo želaného časového obdobia
- zaregistrovanie FCM tokenu a priradenie vozidiel
- odregistrovanie FCM tokenu

Spúšťačom (Trigger) týchto funkcií je komponent Amazon API Gateway, ktorý im zároveň poskytne vstupné údaje. Funkcie následne vykonajú svoju úlohu pri ktorej komunikujú s databázou (viď ďalšia podkapitola) a vrátia volajúcemu späť svoju návratovú hodnotu, čo je v našom prípade JSON dokument so žiadanými dátami z databázy, a/alebo informáciu o výsledku operácie. Funkcia pre príjem nových stavov

vozidiel napríklad dokončí dekompresiu údajov ktorá nebola možná na Sigfox Cloude – konvertovanie dátového typu half späť na float, uloží dokument do databázy, nájde v databáze registráciu FCM tokenu (vysvetlené v podkapitole 4.3.3) k tomuto vozidlu a prepošle dokument na komponent služby push notifikácií (SNS), ktorý ich prostredníctvom FCM doručí na cieľové zariadenie.

4.2.2 Úložiskové a databázové možnosti

Pre ukladanie dát na cloudových platformách je spravidla dostupných viacero možností, pričom účtovanie je spravidla založené na objeme uložených údajov, I/O priepustnosti, využitej prenosovej kapacite a type fyzického úložiska (HDD/SSD disky). Vplyv na cenu má aj to, či ide o vlastné riešenie poskytovateľa cloudových služieb alebo komerčný softvér tretej strany.

4.2.2.1 Relaçné databázy (SQL)

Prvou možnosťou na uloženie údajov sú klasické relačné databázové systémy, postavené na jazyku SQL. AWS ponúka množstvo vlastných aj treťostranných komponentov implementujúcich rôzne RDBMS systémy, ako MySQL, PostgreSQL, MariaDB a ďalšie. Žiaden z nich ale nie je dostupný v rámci always-free Free tier balíka.

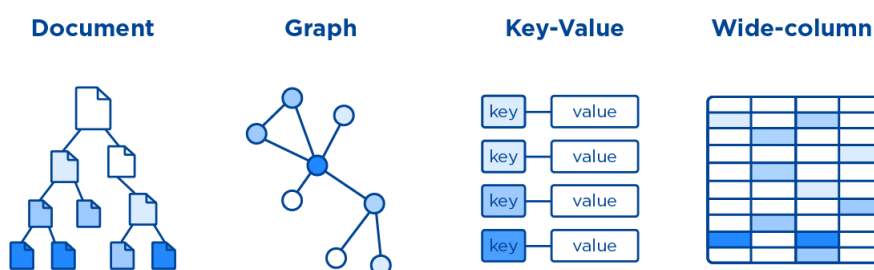
4.2.2.2 Nerelačné databázy (NoSQL)

Nerelačné databázy (označované spoločne ako NoSQL) sú databázové systémy, v ktorých uloženie a spracovanie dát používa iné princípy ako tradičné relačné databázy s tabuľkovými schémami. NoSQL systémy predpísané schémy nepoužívajú a teda v rôznych záznamoch môžu byť uložené rôzne štruktúrované dáta. Motivácia k použitiu iných prístupov môže byť zjednodušenie dizajnu systému, jednoduchá vertikálna aj horizontálna škálovateľnosť a jednoduchšie zabezpečenie dostupnosti. Vďaka inej štruktúre ukladania údajov v porovnaní s RDBMS je aj algoritmická zložitosť pre rôzne operácie iná. V kontexte CAP teorému NoSQL riešenia väčšinou uprednostňujú dostupnosť a odolnosť proti prerušeniu transakcie, pred konzistenciou. Väčšina NoSQL databáz podporuje dopytovací jazyk podobný SQL, no nevýhodou je neexistencia jednotného jazyka pre všetky NoSQL systémy. Ďalšími nevýhodami sú väčšinou absencia plnej podpory ACID transakčného modelu a nižšia úroveň standardizácie.

Všeobecne je možné rozlíšiť 4 základné typy NoSQL databáz [39]:

1. *Dokumentové* – dáta sú ukladané do záznamov nazývaných dokumenty, ktorých obsah môže byť v ľubovoľnom formáte, najčastejšie sa ale používa JSON. Dokumenty sú v databáze adresované pomocou unikátneho kľúča a môžu byť organizované rôznymi spôsobmi – rozdelené do kolekcí, podľa tagov, pomocou skrytých metadát alebo priečinkovou hierarchiou.
2. *Typu kľúč-hodnota* – údaje sú v databáze adresované pomocou kľúčov, ktorých párom môže byť jednoduchý údaj aj komplexný dokument uložený ako JSON. Ide o najjednoduchší typ, v podstate ide o tabuľku s dvomi stĺpcami.
3. *Stĺpcovo-orientované* – údaje sú ukladané do tabuliek, v ktorých ale nemusí mať každý záznam rovnaký počet stĺpcov. Každý záznam má stĺpce definované samostatne, páromi názov-hodnota, ale môžu byť spájané do skupín a agregované pre potreby dopytov. Dajú sa chápať ako dvojrozmerné kľúč-hodnota systémy.
4. *Grafové* – pozostávajú z grafu uzlov a ich prepojení, pričom každý uzol môže ukladať ľubovoľné dáta. Sú vhodné pre ukladanie navzájom zložito súvisiacich údajov, kde sú podstatné nielen dáta, ale aj vzťahy medzi nimi. Súvisiace údaje môžu byť zároveň často obdržané jedinou operáciou.

Tieto typy sa v reálnych riešeniach často prelínajú a vznikajú kombinácie, ktoré sa označujú ako *multimodelové* databázy. Keďže každý typ ponúka iné výhody a nevýhody, vhodnosť použitia konkrétneho typu sa spravidla líši podľa riešeného problému. Segment NoSQL databáz v súčasnosti rastie výrazným tempom, a to najmä v oblastiach spracovania veľkého množstva dát (Big data), webových aplikáciách a v Internete vecí.



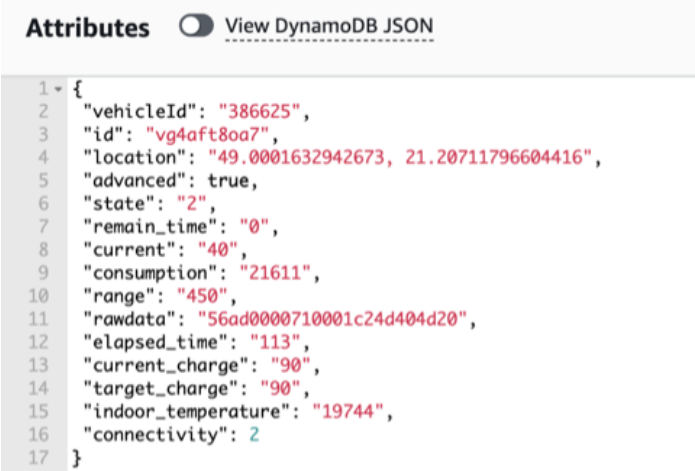
Obr. 4.5: Ilustrácia základných typov NoSQL databáz

AWS ponúka riešenia pre všetky typy, pričom niektoré z nich sú univerzálne a kombinujú podporu viacerých z nich.

4.2.2.3 Súborové úložiská (Buckets)

Poslednou možnosťou sú úložiskové riešenia nazývané “buckets“, do ktorých je možné ukladať dáta vo forme súborov. Tieto úložiská je možné uložiť na uskladnenie súborov potrebných pre iné služby/komponenty, užívateľské dáta, aj verejné súbory, na ktoré sa je možné odkázať klasickou URL adresou. AWS opäť ponúka viacero komponentov, zameraných na mierne odlišné účely použitia.

V našom systéme potrebujeme ukladať iba údaje o registráciách vozidiel k jednotlivým používateľom a prijaté aktualizácie stavov všetkých vozidiel. Na tento účel je najvhodnejšie použitie NoSQL databázy, konkrétne dokumentového typu. Prichádzajúce aktualizácie stavov sú vo formáte JSON, čiže jeden takýto záznam môže jednoducho predstavovať jeden dokument. AWS ponúka viaceré komponenty vlastných aj komerčných dokumentových NoSQL databáz, ako najvhodnejšie pre našu aplikáciu sa javí použitie univerzálnej databázy Amazon DynamoDB, ktorá podporuje ukladanie dokumentových aj key-value dátových štruktúr. V databáze používame 2 tabuľky – jednu pre uloženie stavov vozidiel, v ktorej sú dokumenty particiované (interne triedené) pomocou ID vozidla, a druhú pre uloženie párov používateľ-registrované vozidlo, particiovanú podľa ID používateľa. Pre zjednodušenie (práca nerieši registráciu používateľov do systému a súvisiace záležitosti) sú ale ID používateľov zastúpené priamo FCM tokenmi prislúchajúcimi konkrétnym inštanciam klientskej aplikácie. Nad týmito tabuľkami sú vytvorené indexy, pomocou ktorých je možné vykonávať dopyty na potrebné dáta. Tieto indexy využívajú pri prístupe do databázy Lambda funkcie.



```
Attributes  View DynamoDB JSON
1 {
2   "vehicleId": "386625",
3   "id": "vg4aft8oa7",
4   "location": "49.0001632942673, 21.20711796604416",
5   "advanced": true,
6   "state": "2",
7   "remain_time": "0",
8   "current": "40",
9   "consumption": "21611",
10  "range": "450",
11  "rawdata": "56ad0000710001c24d404d20",
12  "elapsed_time": "113",
13  "current_charge": "90",
14  "target_charge": "90",
15  "indoor_temperature": "19744",
16  "connectivity": 2
17 }
```

Obr. 4.6: Ukážka záznamu stavu vozidla v DynamoDB databáze

4.2.3 API

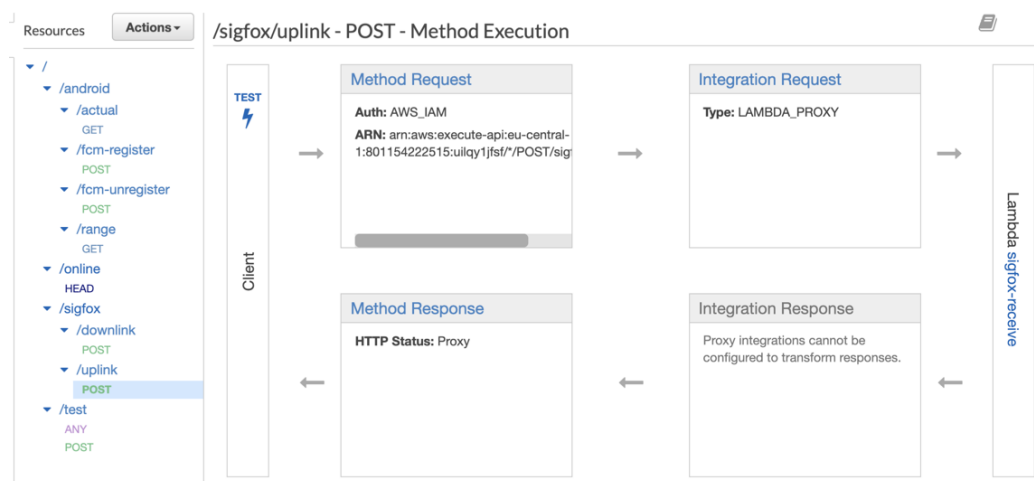
Ako aplikačné rozhranie backendu sme zvolili REST API. Ide o architektúru aplikačného rozhrania, ktorá umožňuje jednoducho vytvárať, čítať, editovať alebo mazať informácie na serveri pomocou jednoduchých HTTP volaní, respektíve je pomocou nich možné vyvolať akcie alebo nastaviť stavy. Každý zdroj (informácia alebo akcia) má svoj vlastný unikátny URI identifikátor a REST definuje metódy pre prístup k týmto zdrojom, ktoré vychádzajú z HTTP metód. Základné z nich, korešpondujúce s CRUD operáciami sú:

- GET – zodpovedá Read – čítanie existujúcej informácie/dát
- POST – zodpovedá Create – vytvorenie novej informácie/odoslanie dát
- PUT – zodpovedá Update – úprava existujúcej informácie
- DELETE – zodpovedá Delete – odstránenie informácie

Okrem toho existuje ďalších 5 menej často používaných HTTP metód: HEAD, TRACE, OPTIONS, CONNECT a PATCH, ktoré je tiež možné v REST rozhraní použiť. Na definíciu API sme použili open-source špecifikáciu OpenAPI, vďaka ktorej je možné API definovať, upravovať a prezentovať samostatne od cieľovej platformy. Ku každej definovanej operácii je možné nastaviť vstupné parametre a ich formát, prípustné odpovede a textové popisy, API ako celku je možné definovať úvodné informácie, verzie, základnú adresu, formát komunikácie a autorizačné metódy.

V našej systéme používame REST API ako rozhranie pre komunikáciu AWS cloudu so všetkými ostatnými časťami systému (Sigfox Cloud, FCM endpoint, používateľská aplikácia). Pre jeho vytvorenie a správu ponúka AWS komponent s názvom Amazon API Gateway, ktorý sme použili. Nové API je možné vytvoriť ručne, alebo importovať OpenAPI špecifikáciu ktorá ho popisuje, zo súboru YAML. My sme použili druhú možnosť, keďže definíciu sme vytvorili samostatne v nástroji Swagger Editor. Časť pre komunikáciu so Sigfox Cloudom sa skladá z metódy pre obdržanie nových správ (na túto metódu sa pripája callback metóda v Sigfox Cloude) a predpripravenej metódy pre downlink komunikáciu. Časť pre komunikáciu s Andorid aplikáciou sa skladá z metód pre získanie posledných známych údajov pre žiadané vozidlo, údajov z určeného časového obdobia pre žiadané vozidlo a metód pre prihlásenie/odhlásenie FCM tokenu (identifikátora zariadenia pre push komunikáciu) na odber aktualizácií konkrétneho

vozidla. Zabezpečenie API je riešené autorizáciou API kľúčom, ktorý je potrebné vložiť medzi hlavičky každého HTTP volania. Grafická podoba definície je priložené v prílohe A. Po zavolaní metód Amazon API Gateway overí autorizáciu volania a po kladnom vyhodnotení spustí príslušné Lambda funkcie, ktoré požiadavku spracujú a prípadne vrátia odpoveď. K tej API Gateway pripojí potrebné hlavičky a stavový kód a pošle ju späť ako odpoveď na príslušné API volanie.



Obr. 4.7: Konfigurácia a diagram API metódy pre príjem aktualizácie

Komponenty AWS ktoré sme v našom systéme použili teda sú:

- API endpoint – Amazon API Gateway
- aplikačný runtime – AWS Lambda
- NoSQL databáza – Amazon DynamoDB
- služba push komunikácie – SNS, Simple Notification Service

Všetky tieto komponenty patria do tzv. always-free Free Tier ponuky, t.j. balíka komponentov, ktoré sú do istého limitu využívania poskytované bezplatne. Tieto limity sú dostatočne vysoké na to aby poskytli slobodu vývoja a prezentácie produktov (prípadne aj ich prevádzkovanie pri malej záťaži), bez obáv z vysokých poplatkov, s ktorými sú často spájané obavy z využívania cloudových služieb. Napríklad limity, ktoré sa týkajú nášho systému sú: 25GB kapacita NoSQL databázy, 1 milión spustení Lambda funkcií mesačne, 1 milión volaní na API Gateway mesačne a 1 milión odoslaných správ cez službu SNS mesačne. Po ich prekročení by sa používanie konkrétneho komponentu začalo automaticky účtovať (po včasnom upozornení) podľa štandardnej cenovej schémy.

4.3 Google Firebase

Google Firebase je platforma, ktorej hlavným účelom je poskytnúť služby pre ľahší a efektívnejší vývoj mobilných a webových aplikácií. Podporované sú platformy Android, iOS a web (JavaScript/Node.js), no pre niektoré funkcie je SDK dostupné aj pre jazyky C++, Java, Python a Go a herný engine Unity. Technicky ide o BaaS (Backend-as-a-Service) koncept, ktorý je súčasťou/podskupinou cloudu Google Cloud Platform (GCP). Firebase poskytuje knižnice a backend pre množstvo často riešených funkcionalít, napríklad cloudová databáza, statický hosting, autentifikácia používateľov, analytika, reklamné služby, A/B testovanie, ale aj pokročilejšie záležitosti, ako strojové učenie alebo presmerovanie používateľa z webu do obsahu aplikácie (deep linking). Služby sú rozdelené do 4 skupín – vývoj, distribúcia a monitorovanie, analytika, a prídavné služby. Všetky služby reálne bežia v GCP, no keďže bol Firebase vyvinutý pre priamu podporu Android platformy, jeho implementácia pri vývoji pre Android je veľmi jednoduchá a je k dispozícii prehľadná dokumentácia. Firebase má tiež vlastné administračné rozhranie, ktoré je oproti rozhraniu GCP jednoduchšie a prehľadnejšie. V prípade potreby je ale možné všetky použité služby manažovať aj priamo v GCP. Rovnako ako v prípade hlavných cloudových platforiem je používanie bezplatné do prekročenia stanovených limitov, ktoré sú nastavené tak aby bolo možné aplikácie pohodlne vyvíjať a prezentovať bez dodatočných nákladov.

Firestore sme použili ako podporu pre klientskú mobilnú aplikáciu, ktorej sa venuje kapitola 5. Z dostupného portfólia sme využili služby pre prihlasovanie používateľov, ukladanie dát a sprostredkovanie push komunikácie.

4.3.1 Authentication

Komponent Authentication poskytuje možnosť jednoducho, no bezpečne implementovať v aplikácii možnosť prihlásenia používateľa, a to pomocou širokej škály spôsobov. Dostupné sú komponenty pre prihlásenie pomocou emailovej adresy a hesla, pomocou protokolov OAuth a OpenID od externých poskytovateľov identity (Google, Apple, Facebook, Twitter, Yahoo, GitHub), pomocou telefónneho čísla a overovacej SMS, pomocou vlastného systému, alebo pre možnosť anonymného prihlásenia. Firebase vedie databázu zaregistrovaných užívateľov, s ktorou požiadavku o prihlásenie danými

údajmi alebo tokenom porovnáva, a poskytne pozitívnu alebo negatívnu odpoveď. K používateľským účtom je tiež možné uložiť niekoľko informácií, ako prezývku alebo adresu k profilovému obrázku. V našej aplikácii sme implementovali prihlasovanie pomocou emailu/hesla a pomocou Google účtu.

4.3.2 Cloud Firestore a Cloud Storage

Cloud Firestore je flexibilná, dokumentovo orientovaná NoSQL databáza, ktorá umožňuje pripojeným aplikáciám ukladať údaje do cloudu a automaticky sa s nimi synchronizovať. Firestore ukladá údaje do dokumentov, ktoré sú rozdelené do kolekcií, prípadne aj subkolekcií. Nad dátami je možné vykonávať pokročilé dopytovanie, distribuovať zmeny na všetky dotknuté zariadenia v reálnom čase, škálovať kapacitu, a veľkou výhodou je aj podpora offline cachovania na klientskej časti, kedy sa zmeny synchronizujú hneď ako sa zariadenie pripojí online. V našej aplikácii sme Firestore použili pre ukladanie informácií o vozidlách daného používateľa, ktoré sa stiahnu po prihlásení do nového zariadenia, alebo sa synchronizujú medzi viacerými jeho zariadeniami.

Cloud Storage je úložisko, ktoré aplikácia môže využiť pre uskladnenie súborov v cloude, alebo kde môžu byť umiestnené externé súbory, ktoré si aplikácia vie stiahnuť. Zabezpečený je robustný prenos dát fungujúci aj cez nekvalitné pripojenie, v prípade výpadku spojenia prenos pokračuje po obnovení tam, kde skončil. Storage je v našej aplikácii použitý pre uloženie obrázkov vozidiel daného používateľa.

4.3.3 Cloud Messaging

Firebase Cloud Messaging (FCM) je systém na prenos push správ, respektíve notifikácií.

4.3.3.1 *Technológia push*

Technológia push správ je princíp internetovej komunikácie, pri ktorej je komunikácia s klientom iniciovaná kvázi zo strany servera. Ide teda o opak štandardne používaného modelu komunikácie typu požiadavka-odpoveď. Interne technológia push využíva buď stále princíp dopytovania sa servera zo strany klienta, kedy sa to však deje neviditeľne

pre služby a aplikácie, ktoré ju využívajú, alebo sa používa TCP spojenie čakajúce v dlhotrvajúcom accept móde [41]. Tak či onak však tento proces zabaľuje vyššia vrstva nad procesmi ktoré technológiu používajú, ktorá zároveň v komunikácii vystupuje ako prijímateľ správ, a sama ich následne distribuuje konkrétnym procesom, ktorým sú finálne určené. V prípade mobilných zariadení túto funkcionality implementuje samotný operačný systém⁵, a teda je nutné aby bola implementovaná priamo jeho vývojárom.

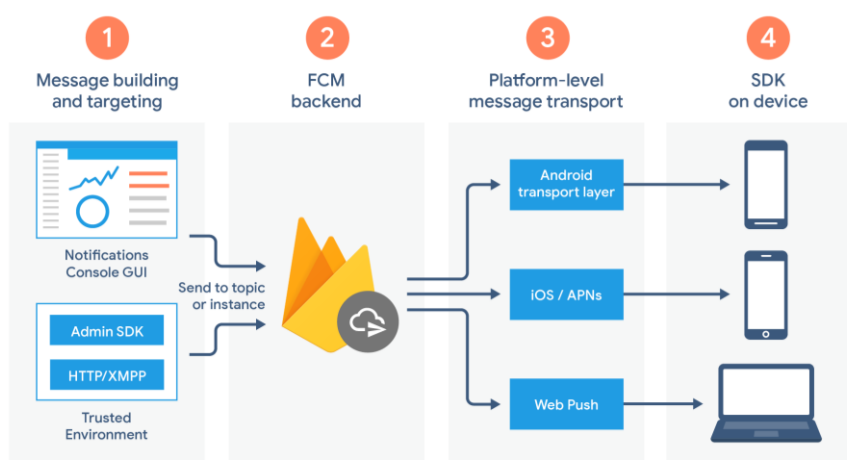
V prípade systému Android je to spoločnosť Google, pričom jej pomenovanie pre implementáciu technológie sa volá Firebase Cloud Messaging (FCM) a funguje ako súčasť platformy Firebase (v prípade systému iOS od spoločnosti Apple sa ich implementácia volá APNs – Apple Push Notification service, pričom FCM vie správy doručovať aj na túto platformu). Hlavnou výhodou a dôvodom vzniku tejto technológie je jej efektívnosť – namiesto toho aby sa x procesov pravidelne dopytovalo rôznych serverov na dostupnosť novej informácie alebo s nimi samo udržiavalo otvorené spojenie (ako to rieši napr. technológia WebSockets), čo je značne energeticky náročné, všetku komunikáciu medzi zapojenými procesmi v zariadení a servermi v internete zabaľuje na jednom konci operačný systém a na druhom konci centrálny cloud, ktorý zbiera a drží všetky správy určené cieľovému zariadeniu. Vytvára sa tým lievrik, kedy operačný systém na základe vlastného uváženia spravuje spojenie, a cloud zasiela správy pre všetkých adresátov na danom zariadení súčasne. Pokiaľ je mobilné zariadenie prebudené, táto komunikácia sa javí ako v reálnom čase, pokiaľ je v uspanom režime, doba doručenia záleží na nastavenej politike. FCM podporuje 2 typy správ – notifikačné a dátové. Prvé z nich sú určené na zobrazenie notifikácií na zariadení používateľa, ktoré ho môžu vyzývať k nejakej akcii alebo informovať o udalosti. Druhé sú určené na prenos malého množstva dát (max 4kB) pre samotnú aplikáciu. Adresátmi môžu byť buď konkrétna inštancia aplikácie (nainštalovaná na nejakom zariadení), skupina inštancií, alebo aplikácie prihlásené na odber kanála (topic) [42,43].

Pre komunikáciu s konkrétnou inštanciou je potrebné aby odosielateľ vopred vedel jej ID, tzv. FCM token. Tento token automaticky vygeneruje lokálne Firebase SDK pri prvom spustení každej aplikácie ktorá FCM implementuje, a je potrebné ho doručiť odosielateľovi. Pre vytvorenie komunikácie odosielateľ pošle správu s tokenom želaného

⁵ Respektíve externá systémová vrstva, v prípade Android je to Google Play Services.

adresáta (adresátov) v JSON formáte na centrálny endpoint na adrese <https://fcm.googleapis.com/fcm/send>, spolu s autorizačným kľúčom v HTTP hlavičke.

Aby sme to docielili v našom systéme, pri každom spustení klientskej aplikácie táto “zaregistruje“ na backende v AWS cloude svoj záujem prijímať aktualizácie o stave daných vozidiel. Používa na to pripravenú metódu REST API (/android/fcm-register), na ktorú odošle svoj pridelený token spolu identifikátormi spravovaných vozidiel. Následne backend v AWS cloude preposiela nové prijaté aktualizácie z týchto vozidiel na FCM backend v GCP cloude, ktorý ich doručí na majiteľove mobilné zariadenie. Služba Firebase Cloud Messaging (rovnako aj Apple Push Notification service) je poskytovaná bezplatne.



Obr. 4.8: Architektúra služby Firebase Cloud Messaging

Firestore ponúka široké možnosti služieb a funkcií pre vývoj moderných mobilných a webových aplikácií, vrátane prostredia Serverless funkcií (FaaS) a databázových aj súborových úložiskových možností. Teoreticky by preto bolo možné celý backend nášho systému hostovať na platforme Firestore, no nie je na takýto účel určená – jej primárnym účelom je podpora aplikácií. Komplexnejšie úlohy alebo ich časti je však možné umiestniť do cloudu GCP s ktorým je Firestore priamo prepojený (je jeho súčasťou). Takéto riešenie by bolo plnohodnotnou alternatívou použitia cloudu AWS.

5 KLIENTSKÁ APLIKÁCIA

Posledným článkom navrhnutého systému je používateľská aplikácia pre smartfóny, vďaka ktorej má majiteľ automobilu prístup k preneseným informáciám a ďalším funkcionalitám. Táto aplikácia predstavuje pre majiteľa vozidla frontend nášho systému, teda všetko s čím prichádza ako používateľ do styku. Ak má systém ako celok plniť úlohu na ktorú bol navrhnutý, teda priniesť majiteľovi elektromobilu zvýšenie komfortu pri jeho používaní, je veľmi dôležité aby bola táto aplikácia navrhnutá premyslene, s dôrazom na dobrý používateľský zážitok (UX). Zlá skúsenosť s aplikáciou by sa totiž preniesla na nespokojnosť s celým systémom a nezaujím o jeho používanie. Kľúčová vlastnosť ktorú užívateľ očakáva, je jednoduchosť používania. Prostredie aplikácie musí byť preto prehľadné a intuitívne.

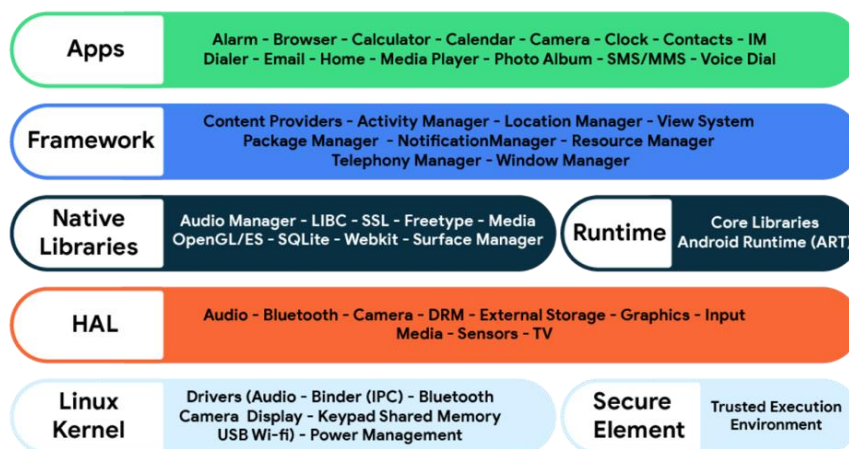
V rámci práce sme si ako cieľovú mobilnú platformu vybrali operačný systém Android a aplikáciu následne vyvinuli natívne pomocou Android SDK a jazyka Java. Funkcie ktoré sme implementovali sú:

1. Prihlasovanie sa používateľským účtom
2. Zobrazenie informácií o priradených vozidlách
3. Prezeranie historických štatistík
4. Zobrazenie polohy a mapy okolia

5.1 Android

Android je mobilní operačný systém založený na modifikovanom jadre Linuxu, ktorý je dostupný ako otvorený softvér (open-source). Je navrhnutý primárne pre zariadenia s dotykovou obrazovkou, teda smartfóny a tablety, no existujú aj verzie pre chytré televízory, hodinky a ďalšie zariadenia. Jeho vývoj vedie spoločnosť Google pod hlavičkou konzorcia firiem Open Handset Alliance a výrobcovia rôznych zariadení môžu Android upravovať pri dodržaní stanovených podmienok. Android má momentálne na trhu mobilných zariadení najväčšie zastúpenie spomedzi všetkých operačných systémov, s podielom 72,2% v apríli 2021 [44]. Pôvodne bol Android predstavený v roku 2007, odvtedy prešiel masívnym vývojom a dnes je na trhu dostupná už jeho 11-ta verzia. Aplikácie pre Android je možné vyvíjať dvoma spôsobmi – buď natívne, v jazyku Java

alebo Kotlin, alebo s použitím frameworkov, ktorých dnes existuje veľké množstvo a spravidla umožňujú vývoj pre viaceré platformy súčasne. Architektúra systému je zobrazená na obrázku nižšie:



Obr. 5.1: Architektúra operačného systému Android

Pre vývojára bežných aplikácií je podstatná vrstva Framework, ktorá pozostáva zo všetkých systémových knižníc potrebných pre napojenie aplikácií na funkcionality systému. Funkcie ktoré tieto knižnice obsahujú sa nazývajú Android API, pričom sú verzované podľa tzv. API levelu, ktorý zodpovedá verziám a podverziám systému Android. API level pre aktuálny Android 11 je 30.

Android definuje 4 základné stavebné komponenty aplikácií:

- Activity – predstavuje viditeľnú časť (jednu obrazovku) aplikácie
- Service – služba, ktorá môže bežať na pozadí nezávisle od aktivít
- Broadcast Receiver – komponent zabezpečujúci komunikáciu údajov a udalostí medzi procesmi aj časťami aplikácie
- Content Provider – poskytovateľ rôznych druhov dát z iných procesov

Ďalším dôležitým komponentom sú tzv. fragmenty, ktoré predstavujú znovupoužiteľné časti UI s vlastným životným cyklom, nezávislým od aktivity ktorá ich hostí. Dôležité je spomenúť ešte mechanizmus zámerov (Intents), čo sú správy, ktoré môžu byť prenášané medzi rôznymi komponentami v rámci aplikácie alebo celého systému, za účelom ich kooperácie na splnení nejakej úlohy. Zámery sa používajú na spúšťanie nových aktivít v

rámci aplikácie, v spolupráci s broadcast-receivermi ako spúšťače metód pri nejakých udalostiach, alebo na komunikáciu s externými aplikáciami alebo systémovými procesmi.

Pre pohodlný vývoj aplikácií pre Android je k dispozícii vlastné IDE s názvom Android Studio, ktoré je postavené na platforme IntelliJ IDEA od spoločnosti JetBrains a ponúka množstvo pokročilých nástrojov, ako UI designer, database inspector, profiler, logcat konzolu, atď., a samozrejme SDK so všetkými potrebnými knižnicami pre jednotlivé verzie systému [45].

5.2 Návrh aplikácie

Pri návrhu a programovaní aplikácie sme dbali na využitie moderných mechanizmov ktoré nám aktuálne Android SDK a Android Studio ponúka, ako napríklad knižnice z balíka Android Jetpack, ktorý poskytuje množstvo komponentov navrhnutých a vyladených pre uľahčenie a skvalitnenie vývoja aplikácií pre Android, a platformy Firebase, ktorá už bola popísaná v kapitole 4.3. Taktiež sme sa držali Googlom odporúčaných návrhových vzorov, ako napríklad single-activity aplikácia, kedy celá aplikácia pozostáva len z jedného komponentu aktivity a jednotlivé obrazovky sú implementované ako fragmenty. V rámci interného rozdelenia funkčných častí aplikácie sme použili model MVVM (Model-View-ViewModel), ktorý logicky oddeľuje grafickú (prezenčnú) časť aplikácie, logiku a zdroje informácií do samostatných celkov.

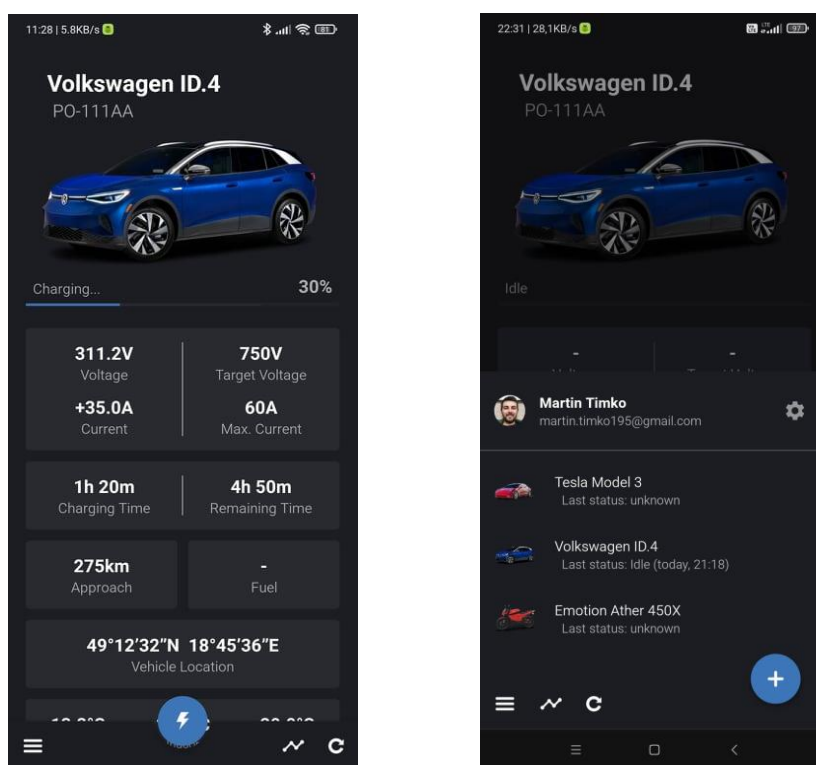
Pri návrhu používateľského rozhrania nám pomohol nástroj designer implementovaný v Android Studiu. Ako GUI komponenty sme použili štandardné Android prvky (views) ako tlačidlá, textviews, editboxy, ale aj dodatočné externé knižnice napr. pre zobrazenie grafov, či mapy. Views sú v rámci GUI rozmiestnené pomocou viacerých rôznych layout komponentov, a to LinearLayout, ConstraintLayout, GridLayout, a moderných CardView a CoordinatorLayout. Pre jednoduchú komunikáciu grafických komponentov s logickou vrstvou aplikácie sme použili mechanizmus View-binding.

Aplikácia pozostáva spolu z 5 obrazoviek: prihlasovacej obrazovky, hlavnej (domovskej) obrazovky, obrazovky histórie, zobrazenia mapy a nastavení.

Prihlasovacia obrazovka sa zobrazí pri prvom spustení aplikácie, resp. po odhlásení používateľa. Prihlásenie je možné pomocou zadania emailovej adresy a hesla, respektíve

tlačidlom Google Sign-In, ktoré zobrazí externý fragment s prihlasovacím formulárom. Po úspešnom prihlásení je používateľ presmerovaný na hlavnú obrazovku.

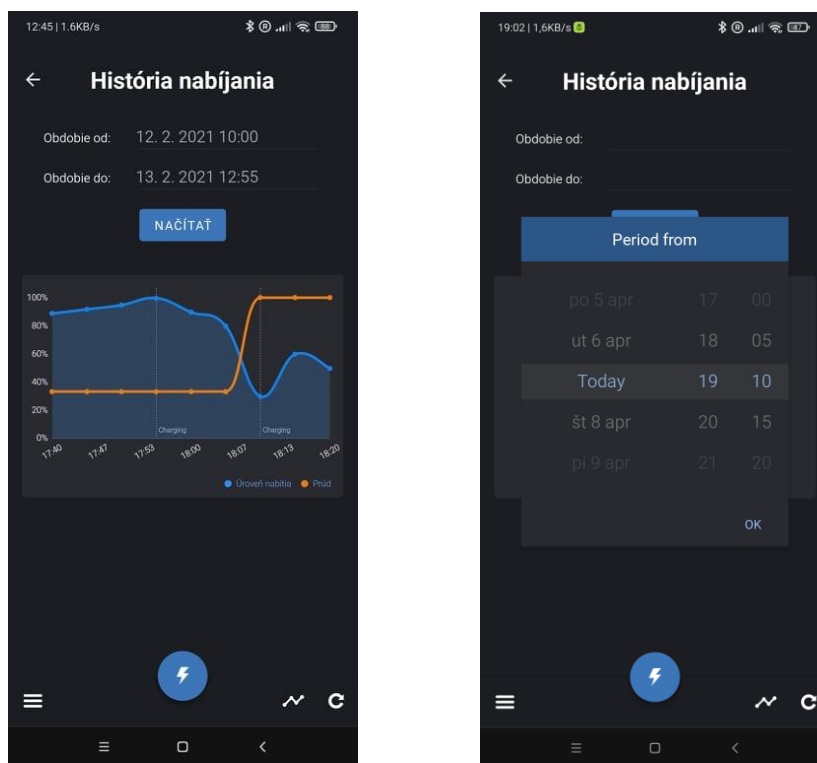
Hlavná obrazovka pozostáva vo vrchnej časti z obrázku zvoleného vozidla a základných informácií o ňom, progressbaru zobrazujúceho aktuálnu a cieľovú úroveň nabitia a dlaždíc s ďalšími informáciami v dolnej časti. Tieto dlaždice sú navrhnuté ako interaktívne, napríklad po kliknutí na polohu sa zobrazí obrazovka s mapovým zobrazením. Na tejto mape je vyznačená poloha všetkých vozidiel daného používateľa a aj poloha samotného mobilného telefónu, ak je dostupná. Na zobrazenie mapy sme použili externú knižnicu Maps SDK for Android. V spodnej časti rozhrania sa nachádza panel s ikonami na ovládanie aplikácie ktorý je možné pohybom nahor vysunúť, čo zobrazí zoznam vozidiel aktuálneho používateľa spolu s ich posledným známym stavom. Z tohto zoznamu je zároveň možné vybrať aktuálne prezerané vozidlo.



Obr. 5.2: Hlavná obrazovka a selektor vozidla

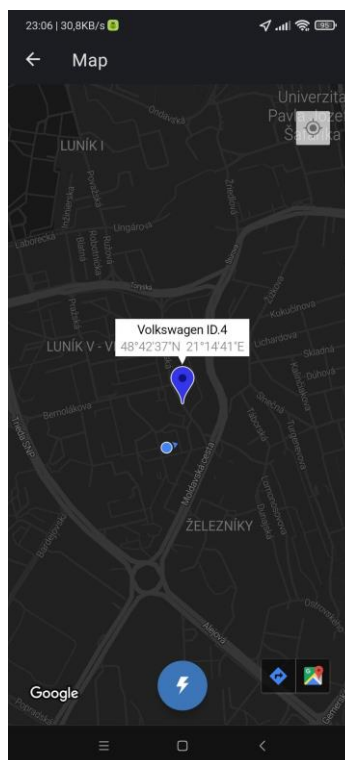
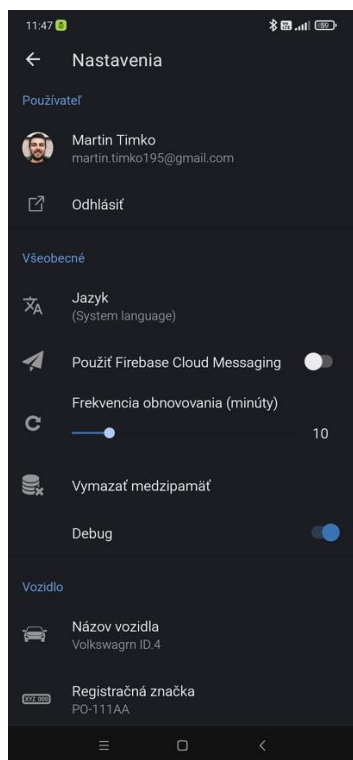
Zo spodného panelu možný prechod na obrazovku histórie, kde sa nachádzajú polia na zadanie časového rozsahu, z ktorých chce používateľ zobrazit' údaje a grafu, kde sú tieto údaje vykreslené. Graf je interaktívny, je možné ho priblížiť, posúvať a vyznačiť' oblasť.

Na jeho zobrazenie sme použili komponent z externej knižnice MPAndroidChart, selektor dátumu je externý komponent SingleDateAndTimePicker. Obe z nich sú dostupné pod licenciou Apache 2.0 [46,47]. Po zvolení želaného časového obdobia sú zobrazené údaje z vyrovnávacej pamäte (lokálnej databázy) a zároveň je odoslaná požiadavka na API backendu (metóda /android/range). O priebehu sieťovej požiadavky informuje načítavací prúžok, po prijatí odpovede sú nové dáta do zobrazeného grafu doplnené.



Obr. 5.3: Obrazovka histórie a selektor dátumu a času

Poslednou obrazovkou prístupnou pomocou ikony v spodnom vysúvacom paneli sú nastavenia aplikácie, kde sa nachádzajú možnosti na zmenu prezývky používateľského účtu, odhlásenie sa z účtu, zmenu jazyka aplikácie (je dostupná lokalizácia do slovenčiny a angličtiny), zmenu niekoľkých preferencií správania sa aplikácie, nastavenie názvu, obrázka a údajov aktuálneho vozidla, a niekoľko informácií o aplikácii. Táto obrazovka je na rozdiel od ostatných vytvorená pomocou komponentu PreferenceScreen a jeho podkomponentov pre jednotlivé typy nastavení – EditTextPreference, SwitchPreference atď. Zvolené nastavenia sú ukladané do úložiska SharedPreferences a lokálnej databázy, do tabuľky vozidiel.



Obr. 5.4: Obrazovky nastavení a zobrazenia mapy

Prihlasovanie je riešené pomocou služby Firebase Authentication. Po prihlásení sa z cloudovej databázy Firebase Firestore stiahnu údaje daného používateľa o jeho spárovaných dopravných prostriedkoch a ich obrázky z Firebase Cloud Storage a sú uložené do lokálnej databázy. Následne si po prihlásení (a každom ďalšom spustení aplikácie) aplikácia vyžiada z backendu najaktuálnejšie údaje o zvolenom vozidle, a to prostredníctvom HTTP požiadavky na REST API (metóda /android/actual). Taktiež sa po prihlásení (a preventívne aj po každom spustení aplikácie, keďže sa môže zmeniť), ak je to povolené v nastaveniach, odošle na backend FCM token spolu so zoznamom príslušných vozidiel (metóda /android/fcm-register). V závislosti od nastavenia v preferenciách aplikácie sú ďalšie updaty stavu vozidla prijímané už pomocou mechanizmu push správ (viď kapitola 4.3.3), alebo sa na ne aplikácia periodicky dopytuje na REST API rovnakou metódou ako prvýkrát po spustení. Vždy po prijatí nového stavu ľubovoľného vozidla je tento stav najprv uložený do lokálnej databázy aplikácie a následne, ak ide o aktuálne zvolené vozidlo, sú prekreslené dlaždice s informáciami, prípadne iné prvky. Lokálna databáza je ako medziprvok použitá z dvoch dôvodov. Po prvé, aplikácia kontroluje hneď po spustení, či sa v nej nenachádza aktuálny stav vozidla.

V prípade že posledný záznam nie je starší ako nastavená hranica, hneď ho zobrazí, čo šetrí čas načítavania aktuálnych údajov zo siete po každom spustení aplikácie. Po druhé, je použitá ako prvotný zdroj údajov pri požiadavke na vykreslenie grafu na obrazovke história, až následne sú údaje prípadne doplnené ďalšími, prijatými z backendu. To je týmto spôsobom implementované z rovnakého dôvodu (rýchlejšie načítavanie). V prípade, že úroveň nabitia dosiahne cieľovú hodnotu (ukončenie nabíjania), aplikácia zobrazí v systéme notifikáciu.

5.3 Komunikácia

HTTP komunikácia s REST API backendu je implementovaná s pomocou obľúbenej knižnice Retrofit. Ide o open-source knižnicu pre Android, ktorá je nadstavbou nad Java HTTP klientom OkHttp. Retrofit pridáva funkciu automatickej deserializácie dát do objektov a implementovania cieľového API ako Java rozhrania s volateľnými metódami. Vytvorené rozhranie (Java interface) môže mať teda rovnakú podobu, ako definícia samotného REST API pomocou OpenAPI špecifikácie.

Komunikácia pomocou push správ je implementovaná pomocou Firebase SDK, kedy je po prijatí novej správy zavolaná callback metóda v samostatnom komponente služby. Tá prijatý JSON dokument zparsuje, payload deserializuje a vytvorený objekt zabalený do intentu pomocou broadcast mechanizmu odošle do hlavnej aktivity, kde je ďalej spracovaný.

5.4 Ukladanie dát

Prijaté aktualizácie stavov vozidiel, ako aj informácie o registrovaných vozidlách sú ukladané do lokálnej relačnej SQLite databázy aplikácie. Na prácu s ňou sme použili knižnicu Room, ktorá je súčasťou balíka Android Jetpack. Room poskytuje podobne ako Retrofit funkciu implementácie databázy ako Java rozhrania s volateľnými metódami a možnosť automatickej serializácie a deserializácie objektov tak, aby boli uložitelné do databázy, keďže SQLite podporuje iba dátové typy TEXT, INTEGER, REAL a BLOB. V databáze dáta ukladáme do dvoch tabuliek – jednej pre vozidlá a druhej pre stavy vozidiel. Obsah tabuľky vozidiel je synchronizovaný s cloudovou databázou Firestore.

id	vehicleId	timestamp	connec	state	current_charge	target_charge	current	elapsed_time	remain_time	range
8a9cb3207d	386625	1619417576	2	1	59	90	120	29	103	299
feca02f337	386625	1619417332	2	1	60	90	120	30	102	302
23d2bfb66f	386625	1619417354	2	1	70	90	120	60	81	351
719bbbfdb5	386625	1619417597	2	1	70	90	120	60	81	351
a782fa7e1e	386625	1619417375	2	1	77	90	120	91	59	389
a4ee0d9ed7	386625	1619417620	2	1	77	90	120	90	59	389
dd93abb092	386625	1619416388	2	1	83	90	120	121	37	417
b76524d72d	386625	1619417641	2	1	83	90	120	121	37	417
a01d19bc98	386625	1619417663	2	1	87	90	120	152	15	438
764090d6c9	386625	1619417679	2	2	90	90	0	175	0	450

Obr. 5.5: Výrez dát z tabuľky stavov vozidiel

Preferencie z obrazovky nastavení a niekoľko ďalších informácií (FCM token, ID aktuálne/posledne vybraného vozidla, cesta k ikone používateľa, ...) sú ukladané do úložiska SharedPreferences, čo je jednoduché úložisko aplikácie typu kľúč-hodnota, do ktorého poskytuje prístup samotný android framework. Uložené údaje môžu byť len dátového typu boolean, int, long, float alebo string.

```

{
  debug: 
  fcm_enabled: 
  fcm_token: "efrfnxCTl0-qfYlSbSCmy:APA91bEmEE6h7QzmQ5Q1kSduGX8bePxEazHLAK3-
0fSnKlTh0SeBIzaPtKBV2t-
x6pS7ZkLGfb_wGkHwhe0FyoWLS7nLzXXZcnkrFfo5RIZwTqPHwRK3la9izHgHH5RQvNQmsuECuMZ"
  language: "en_US"
  last_vehicle_id: "386625"
  update_interval: 5
  user_icon: "usericon_1035221057196117284.png"
}

```

Obr. 5.6: Údaje ukladané do SharedPreferences

Posledným typom dát ktoré je treba ukladať, sú obrázky vozidiel a profilový obrázok používateľa. Tie sú ukladané ako súbory do interného úložiska telefónu, do priečinku inštalácie aplikácie (App-specific – Internal storage), ku ktorému má prístup len samotná aplikácia a systém.

Ďalšie funkcie na ktoré je aplikácia pripravená, no neimplementovali sme ich sú: možnosti vzdialeného ovládania vozidla (ovládanie vykurovania/klimatizácie, zmena nabíjacieho prúdu a cieľovej úrovne nabitia), prispôbenie pre rôzne typy dopravných prostriedkov a proces registrácie nových vozidiel.

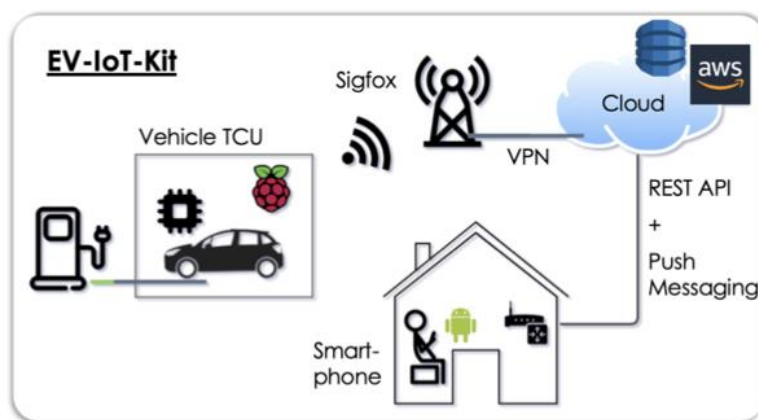
6 ZÁVER

Hlavným cieľom práce bolo navrhnuť a vytvoriť systém, ktorým môže elektromobil alebo iné vozidlo komunikovať s aplikáciou v chytrom telefóne svojho majiteľa. Majiteľ moderného elektromobilu by od takejto služby prirodzene očakával 3 vlastnosti:

1. Jednoduchosť
2. Dostupnosť
3. Spoľahlivosť

Jednoduchosť je kľúčová pre ochotu majiteľa vozidla tento systém používať – musí mu priniesť zvýšenie komfortu. Dostupnosť znamená, že systém bude funkčný vždy vtedy, keď ho používateľ bude chcieť použiť – bez ohľadu na to či sa nachádza doma, na cestách, alebo v zahraničí. Spoľahlivosť súvisí s predchádzajúcim bodom, no rozumieme pod ňou to, že systém bude spoľahlivo a automaticky fungovať aj bez potrebných zásahov používateľa, aj v neoptimálnych podmienkach – napríklad v oblastiach s horším pokrytím signálu alebo v zarušenom prostredí.

Systém, ktorý sme navrhli všetky tieto požiadavky spĺňa. Práca po kapitolách postupne prechádza jeho jednotlivé časti. Prvá kapitola uvádza čitateľa do problematiky. Druhá sa venuje simulátoru telematickej jednotky, ktorá by bola v reálnom riešení integrovaná do vozidla využívajúceho službu, prípadne inštalovaná ako samostatné zariadenie. Tretia kapitola sa venuje prenosovej časti medzi vozidlom a backendom služby. Štvrtá kapitola rozoberá samotný backend služby, jeho umiestnenie do verejného cloudu Amazon Web Services a využitie ďalších cloudov. Posledná piata kapitola je o vytvorení mobilnej klientskej aplikácie pre OS Android.



Obr. 6.1: Architektúra navrhnutého systému a použité technológie

Teoretická časť práce sa venovala vysvetleniu trendov a konceptov moderných technológií, ako sú Internet of Things, smart city a cloud. Najobsiahlejšou časťou je kapitola o bezdrôtových technológiách ktorými je možné zabezpečiť pripojenie IoT zariadení na internet, a ich porovnanie. Dôraz sme kládli na technológie typu LPWAN, keďže ich vlastnosti najviac zodpovedajú trom spomínaným požiadavkám, ktoré sú na náš systém kladené. Ako primárnu komunikačnú technológiu sme zvolili Sigfox, vďaka ktorému je konektivita možná na veľké vzdialenosti od základňových staníc, siete sú, alebo v blízkej budúcnosti budú dostupné vo veľkom množstve krajín po celom svete, a koncové zariadenie môže medzi týmito sieťami plynulo prechádzať bez potreby riešiť medzinárodný roaming.

Navrhnutý systém však nie je konečným riešením určeným na priame použitie v praxi. Je to skôr demonštračné riešenie a základová platforma, ktorej časti sú navrhnuté modulárne tak, že ak by niektorá z nich nevyhovovala požiadavkám konkrétnej konečnej aplikácie, je jednoduché ju vymeniť za inú – napríklad použiť inú prenosovú technológiu alebo umiestniť logiku do cloudu iného poskytovateľa, prípadne do svojho vlastného. Navrhnuté riešenie nemusí slúžiť len elektromobilom alebo automobilom – mobilnú aplikáciu je možné jednoducho pozmeniť tak, aby bolo možné do systému zapojiť aj iné dopravné prostriedky, napríklad motorky, skútre, bicykle a elektrické kolobežky. Používateľ by tak mohol všetky svoje vlastnené dopravné prostriedky spravovať v jednej univerzálnej aplikácii. Iným príkladom použitia môže byť sharingová služba na zdieľanie ktoréhokoľvek z týchto dopravných prostriedkov. Nahradením dopravného prostriedku ako koncového zariadenia iným typom zariadení môže byť riešenie použité ako základ pre množstvo rôznych aplikácií napríklad aj v priemyselnom prostredí, kde problematika prechádza do oblasti Štvrtej priemyselnej revolúcie, v ktorej je využitie IIoT základnou myšlienkou.

Ako celok môže práca teda slúžiť ako dobrý príklad moderného IoT riešenia nejakého problému a pomôcť ako sprievodca, z čoho všetkého sa takýto systém skladá, ako jednotlivé časti spolupracujú a ako postupovať pri jeho návrhu. Obzvlášť prínosné to môže byť v súčasnej dobe, kedy je svet IoT riešení a cloudových technológií na prudkom vzostupe, no zároveň sú tieto oblasti tak rozsiahle, že vývojár alebo firma, ktorá by chcela pracovať na vývoji nejakého produktu, no nemá v tejto oblasti skúsenosť, potrebuje venovať veľmi veľké množstvo času len na zorientovanie sa v problematike.

ZOZNAM POUŽITEJ LITERATÚRY

- [1] KASAJ, O. (2018). Autá budúcnosti na dosah. [online] URL: <https://www.globallogic.com/sk/blog/auta-buducnosti-na-dosah/>
- [2] <https://digital-strategy.ec.europa.eu/en/news/ecall-all-new-cars-april-2018>
- [3] GLOBALLOGIC S.R.O. (2019). Budúcnosť áut máme vo svojich rukách [online] URL: <https://www.globallogic.com/sk/blog/buducnost-aut-mame-vo-svojich-rukach/>
- [4] RASPBERRY PI FOUNDATION. Raspberry Pi 4 Computer Overview [online] URL: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf>
- [5] PARMAR, M. (2020). Windows 10 now runs on the Raspberry Pi 4 and 3 [online] URL: <https://www.windowslatest.com/2020/02/09/heres-how-windows-10-runs-on-raspberry-pi-4-and-3/>
- [6] LPWAN CO., Ltd. Iottoall Sigfox Devkit. [online] URL: <https://www.gme.sk/data/attachments/dsh.773-011.2.pdf>
- [7] BOSCH. BMP280 Digital Pressure Sensor [online] URL: <https://www.gme.sk/data/attachments/dsh.772-274.2.pdf>
- [8] Standard Range Plus Supercharging Speed [online] URL: <https://teslamotorsclub.com/tmc/threads/standard-range-plus-supercharging-speed.146816/page-9>
- [9] The Institute of Electrical and Electronics Engineers (2008). IEEE 754-2008 [online] URL: <https://standards.ieee.org/standard/754-2008.html>
- [10] LAMBERT, F. (2018). Tesla releases some of its software to comply with open source licences [online] URL: <https://electrek.co/2018/05/19/tesla-releases-software-open-source-licences/>
- [11] WIKIPEDIA.ORG. Qt (software) [online] URL: [https://en.wikipedia.org/wiki/Qt_\(software\)](https://en.wikipedia.org/wiki/Qt_(software))
- [12] BEHRTECH. LPWAN Technologies Comparison #2 [online] URL: <https://behrtech.com/blog/lpwan-technologies-comparison-2/>
- [13] AMR.SK. Technológia NB-IoT (Narrow Band) [online] URL: <http://amr.sk/?s=tech-nbiot>
- [14] TREND.SK (2019) Slovenskí operátori zabrali, siete pre Internet vecí výrazne narástli [online] URL: <https://www.trend.sk/trend-archiv/slovenski-operatori-zabrali-siete-pre-internet-veci-vyrazne-narastli>
- [15] PEŠKA, R. (2019). LTE-M v otázkách a odpovediach [online] URL: <https://vyvoj.hw.cz/lte-m-v-otazkach-a-odpovedich.html>

- [16] ETSI. (2018). ETSI EN 300 220-2 [online] URL:
https://www.etsi.org/deliver/etsi_en/300200_300299/30022002/03.02.01_60/en_30022002v030201p.pdf
- [17] SONNENBERG, J. FCC Part 15 ISM Regulations [online] URL:
<https://www.raveon.com/wp-content/uploads/2019/05/AN203FCC-ISM.pdf>
- [18] THETHINGS NETWORK (2017). Ground breaking world record! LoRaWAN packet received at 702 km (436 miles) distance [online] URL:
<https://www.thethingsnetwork.org/article/ground-breaking-world-record-lorawan-packet-received-at-702-km-436-miles-distance>
- [19] THETHINGS NETWORK (2021). Limitations: data rate, packet size, 30 seconds uplink and 10 messages downlink per day Fair Access Policy [guidelines] [online] URL: <https://www.thethingsnetwork.org/forum/t/limitations-data-rate-packet-size-30-seconds-uplink-and-10-messages-downlink-per-day-fair-access-policy-guidelines/1300>
- [20] KM TECHNOLOGY, SPOL. Zabezpečenie LoRaWAN sietí [online] URL:
<https://km-technology.sk/sk/blog/zabezpecenie-lorawan-sieti>
- [21] TELEMATIC. webstránky, URL: <https://telemetric.tech/en/lorawan-technology-and-specifications/>
- [22] HWANG, Y. (2020) Cellular IoT Explained - NB-IoT vs. LTE-M vs. 5G and More [online] URL: <https://www.iotforall.com/cellular-iot-explained-nb-iot-vs-lte-m>
- [23] NORDRUM, A. (2017). Everything You Need to Know About 5G [online] URL: <https://spectrum.ieee.org/video/telecom/wireless/everything-you-need-to-know-about-5g>
- [24] 3GPP. webstránky. URL: <https://www.3gpp.org/specifications/67-releases>
- [25]] WIKIPEDIA.ORG. IEEE 802.11[online] URL:
https://en.wikipedia.org/wiki/IEEE_802.11
- [26] SIGFOX (2020). Connected Objects Radio Specifications, rev 1.5
- [27] SIGFOX. Sigfox Device Radio Specifications [online] URL:
<https://build.sigfox.com/sigfox-device-radio-specifications>
- [28] GOMEZ, C. (2019) A Sigfox Energy Consumption Model [online] URL:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6387435/>
- [29] SIGFOX (2021). Introducing 0G network [online] URL:
https://www.sigfox.com/sites/default/files/og-guide/Sigfox%20-%20Introducing%200G_Jan2021.pdf
- [30] SIGFOX. Radio Configurations [online] URL: <https://build.sigfox.com/sigfox-radio-configurations-rc#geographical-zones>
- [31] SIGFOX. Coverage information [online] URL:
<https://www.sigfox.com/en/coverage>

- [32] SIGFOX SLOVAKIA (2020). Aktuálne pokrytie Sigfox na Slovensku [online] URL: https://sigfoxslovakia.com/frontpage11/simplecell_sigfox_pokrytie_mapa/
- [33] <https://twitter.com/MyHidnSeek/status/832248324463984640>
- [34] SIGFOX. Geolocation technologies [online] URL: <https://build.sigfox.com/geolocation-technologies#sigfox-network-location>
- [35] NICOLAS CHUBERRE. Satellite components for the 5G system [online] URL: https://www.3gpp.org/news-events/1933-sat_ntn
- [36] MICROSOFT. Principles of cloud computing [online] URL: <https://docs.microsoft.com/en-us/learn/modules/principles-cloud-computing>
- [37] RICHTER, F. (2021). Amazon Leads \$130-Billion Cloud Market [online] URL: <https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/>
- [38] ORANGE. Orange IoT Connectivity as a powerful driving force [online] URL: <https://www.allthingstalk.com/en/orange/iot/what-is-orange-iot>
- [39] MongoDB. Understanding the Different Types of NoSQL Databases [online] URL: <https://www.mongodb.com/scale/types-of-nosql-databases>
- [40] Aira GROUP, s.r.o., Co je cloud computing [online] URL: <https://www.sprava-site.eu/cloud-computing/>
- [41] STACKOVERFLOW. How does push notification technology work on Android? URL: <https://stackoverflow.com/questions/11508613/how-does-push-notification-technology-work-on-android>
- [42] KAMATH A. (2020). What are Push Notifications? An In-depth 2020 Guide [online] URL: <https://www.moengage.com/blog/what-are-push-notifications/>
- [43] WIKIPEDIA.ORG. Push Technology [online] URL: https://en.wikipedia.org/wiki/Push_technology
- [44] Mobile Operating System Market Share Worldwide (April 2021) URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- [45] WIKIPEDIA.ORG. Android [online] URL: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- [46] PhilippJahoda. MPAndroidChart. URL: <https://github.com/PhilJay/MPAndroidChart>
- [47] Florent Champigny. SingleDateAndTimePicker. URL: <https://github.com/florent37/SingleDateAndTimePicker>

ČESTNÉ VYHLÁSENIE

Vyhlasujem, že som zadanú diplomovú prácu vypracoval samostatne, pod odborným vedením vedúceho diplomovej práce Róberta Hudeca, prof. Ing. PhD. a používal som len poznatky získané počas štúdia a zdroje uvedené v práci.

V Košiciach, dňa 5. mája 2021



Bc. Martin Timko

PRÍLOHOVÁ ČASŤ

ZOZNAM PRÍLOH

Príloha A: Vizuálna forma OpenAPI definície aplikačného rozhrania backendu služby

Príloha B: Zdrojový kód simulátora telematickej jednotky

Príloha C: Zdrojový kód Lambda funkcií z AWS cloudu

Príloha D: Zdrojový kód klientskej aplikácie pre Android

Všetky prílohy sa nachádzajú na priloženom CD nosiči.