



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

APLIKACE PRO SIMULACI A VIZUALIZACI KOMUNIKACE V MOBILNÍCH SÍTÍCH 4G/5G DLE 3GPP.

APPLICATION FOR COMMUNICATION SIMULATION AND VISUALISATION IN 3GPP 4G/5G MOBILE NETWORKS.

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jozef Fašanga

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Radim Dvořák

BRNO 2024

Diplomová práce

magisterský navazující studijní program **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Jozef Fašanga

ID: 220812

Ročník: 2

Akademický rok: 2023/24

NÁZEV TÉMATU:

Aplikace pro simulaci a vizualizaci komunikace v mobilních sítích 4G/5G dle 3GPP.

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je vytvoření aplikace pro simulaci komunikace jednotlivých prvků v mobilních sítích na úrovni L3 a výše. Aplikace bude schopna simulovat běžné procedury týkající se zejména připojení a odpojení zařízení do mobilní sítě, jak z pohledu zařízení, tak z pohledu mobilní sítě. Průběh komunikace bude uživateli vizuálně zobrazován a uživatel bude mít možnost ve vytvořeném grafickém rozhraní měnit jednotlivé parametry sítě a zařízení, případně komunikaci "krokovat". V teoretické části práce bude detailně rozebrána struktura mobilní sítě páté generace (5G), včetně popisu jednotlivých prvků, jejich funkcionality a rozhraní mezi nimi. K výše uvedenému bude také popsán přenosový model a důležité mechanismy/protokoly pro funkcionality mobilní sítě. Následovat bude rozbor vybraných scénářů komunikace dle 3GPP standardu. V praktické části dojde k implementaci samotné aplikace, kde důraz bude kladen na správnost implementace z hlediska standardu 3GPP, vizualizaci a možnosti uživatelské konfigurace pro jednotlivé scénáře.

DOPORUČENÁ LITERATURA:

Dle pokynů vedoucího práce.

Termín zadání: 5.2.2024

Termín odevzdání: 21.5.2024

Vedoucí práce: Ing. Radim Dvořák

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Mobilné siete formujú kľúčový pilier súčasného života, umožňujúc pripojenie bez ohľadu na miesto. Práca sa zameriava na vývoj a simuláciu mobilných sietí 4G/5G podľa štandardu 3GPP. Siete 4G/5G prešli vývojom od jednoduchých hlasových služieb k sofistikovaným systémom pre prenos dát. Simulačné nástroje zjednodušujú pochopenie ich komplexnosti. Práca ponúka pohľad na históriu a budúcnosť týchto sietí, pričom popisuje procesy návrhu aplikácií z hľadiska dizajnéra aj vývojára. Teoretická časť sa zaoberá analýzou sietí 4G/5G, ich architektúrou a novými technológiami. Záverečná časť zdôrazňuje oblasť grafického návrhu aplikácií so zreteľom na estetiku a užívateľskú prívetivosť. Práca ďalej popisuje návrh samotnej aplikácie, spolu s vysvetlením správnej voľby programovacieho jazyka a nástrojov na realizáciu.

KLÚČOVÉ SLOVÁ

mobilné siete, 4g, 5g, 3gpp štandard, evolúcia sietí, prenos dát, simulačné nástroje, návrh aplikácií, užívateľská skúsenosť (ux), užívateľský dizajn (ui), grafický dizajn, architektúra sietí, pripojenie, analýza technológií, budúce trendy, estetika, web, react, formik

ABSTRACT

Mobile networks constitute a key pillar of contemporary life, enabling connectivity regardless of location. This work focuses on the development and simulation of 4G/5G mobile networks according to the 3GPP standard. 4G/5G networks have evolved from simple voice services to sophisticated systems for data transmission. Simulation tools simplify the understanding of their complexity. The paper provides insight into the history and future of these networks, describing application design processes from both the designer's and developer's perspectives. The theoretical section analyzes the architecture and new technologies of 4G/5G networks. The final part emphasizes the field of graphic design for applications with a focus on aesthetics and user-friendliness. The work further describes the design of the application itself, along with an explanation of the correct choice of programming language and tools for implementation.

KEYWORDS

mobile networks, 4G, 5G, 3GPP standard, network evolution, data transmission, simulation tools, application design, user experience (UX), user interface (UI), graphic design, network architecture, connectivity, technology analysis, future trends, aesthetics, web, react, formik

FAŠANGA, Jozef. *Aplikace pro simulaci a vizualizaci komunikace v mobilních sítích 4G/5G dle 3GPP.* Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2023. Vedúci práce: Ing. Radim Dvořák

Vyhlásenie autora o pôvodnosti diela

Meno a priezvisko autora: Bc. Jozef Fašanga
VUT ID autora: 220812
Typ práce: Diplomová práca
Akademický rok: 2023/24
Téma záverečnej práce: Aplikace pro simulaci a vizualizaci komunikace v mobilních sítích 4G/5G dle 3GPP.

Vyhlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedúcej/cého záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora*

*Autor podpisuje iba v tlačenej verzii.

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce pánovi Ing. Radimovi Dvořákovi za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Obsah

Úvod	11
1 Mobilné siete	12
1.1 História mobilných sietí	12
1.2 Frekvenčné spektrum	13
1.2.1 Rozdelenie spektra v rámci mobilných technológií	14
1.3 Štandardizácia mobilných technológií	15
1.3.1 3GPP	16
1.3.2 Spätná kompatibilita	16
1.4 3GPP Release	16
2 Technológie 4G	18
2.1 Základné princípy 4G	18
2.2 LTE	18
2.3 Architektúra 4G	19
3 Technológie 5G	23
3.1 Mobilná sieť ako ekosystém	23
3.2 Architektúra 5G	25
3.2.1 Standalone a Non-Standalone nasadenia	25
3.2.2 Referenčný model architektúry	27
3.2.3 Model architektúry NSA	28
3.3 Porovnanie oproti staršej generácii - 4G	28
3.3.1 Zvýšená šírka pásma	28
3.3.2 Masívne MIMO	29
3.3.3 MIMO - Beam forming	30
3.3.4 Nižšia latencia	31
4 Procesy v mobilných sieťach	33
4.1 GTP protokol	33
4.2 Pripojenie k sieti	33
4.3 Odpojenie od sieti	36
5 Voľba nástrojov na realizáciu aplikácie	38
5.1 Nástroje potrebné pre beh aplikácie	38
5.1.1 Kontajnerová virtualizácia	38
5.1.2 Kompilátor	39
5.1.3 Správca balíčkov	39

5.2	Vývojové nástroje	40
5.2.1	Vývojové prostredie	40
5.2.2	Lintre a formáteri	40
5.2.3	Git a GitHub	41
5.3	Programovacie jazyky + Frameworky	41
5.3.1	Frontend aplikácie	41
5.3.2	React + TypeScript	42
5.3.3	Štýly aplikácie	44
5.3.4	Frameworky	46
5.3.5	Backend aplikácie	52
5.4	Grafické rozhranie	52
5.4.1	UX a UI dizajn	52
5.4.2	Programy na návrh UI dizajnu	52
6	Aplikácia pre simuláciu scenárov v mobilných sieťach	54
6.1	Obecná štruktúra	54
6.2	Dizajn UI a funkcionality	55
6.3	Nastavenia prvkov	60
	Záver	61
	Literatúra	62
	Zoznam symbolov a skratiek	66
	Zoznam príloh	69
	A Aplikácia	70
	B Návrh UI aplikácie	71
	C UI história správ	72
	D UI detail správy	73
	E UI modal okno	74
	F Komponenty UI 1	75
	G Komponenty UI 2	76
	H Elektronická príloha	77

Zoznam obrázkov

1.1	Frekvenčné pásma - graf	14
1.2	Frekvenčné pásma	15
2.1	Architektúra 4G	19
2.2	Porovnanie úloh E-UTRAN a EPC	22
3.1	Možnosti využitia 5G	24
3.2	Porovnanie Standalone a Non-Standalone	26
3.3	5G Non-roaming architektúra	27
3.4	5G NSA architektúra	28
3.5	Šírka pásma	29
3.6	MIMO anténa	30
3.7	MIMO Beam forming	31
3.8	Rozdelená užívateľská a riadiaca rovina	32
4.1	Pripojenie k sieti - graf	36
4.2	Odpojenie od sieti - graf	37
5.1	Docker kontajnery	39
5.2	Nástroje Figma	53
6.1	Vývojový diagram aplikácie	55
6.2	Grafický návrh aplikácie	56
6.3	Ovládacie prvky aplikácie	56
6.4	Nastavenia aplikácie	57
6.5	Nápoveda aplikácie	57
6.6	Komponent architektúry v aplikácii	58
6.7	Detail správy	58
6.8	Modal okno	59
6.9	Upozornenia	59
A.1	Aplikácia	70
B.1	Grafický návrh aplikácie	71
C.1	Grafický návrh aplikácie - história správ	72
D.1	Grafický návrh aplikácie - detail správy	73
E.1	Grafický návrh aplikácie - modal okno	74
F.1	Komponenty aplikácie 1	75
G.1	Komponenty aplikácie 2	76

Zoznam výpisov

5.1	React komponent	42
5.2	React typy	43
5.3	Použitie komponentu	43
5.4	Použitie hooku	43
5.5	Vytvorenie kontextu	44
5.6	Pužitie kontextu	44
5.7	Príklad použitia BEM	45
5.8	Použitie zod	46
5.9	Použitie zod	46
5.10	Použitie formiku	47
5.11	Použitie FormikForm	47
5.12	Vytvorenie use-between	48
5.13	Použitie use-between	48
5.14	Konfigurácia TanStack Query	49
5.15	Použitie TanStack Query Provider	49
5.16	Použitie hooku useQuery	49
5.17	Použitie hooku useMutation	50
5.18	Invalidovanie query	50
5.19	React Flow Node definícia	51
5.20	React Flow Edge definícia	51

Úvod

Mobilné siete tvoria kľúčový pilier súčasnej spoločnosti, kde koncept pripojenia nie je viazaný na pevné miesto. Od svojho vzniku v minulých generáciách sa mobilné siete vyvíjali z jednoduchých hlasových služieb na komplexné a vysoko výkonné systémy pre prenos dát. Komplexnosť dnešných sietí a veľké množstvo informácií, súvisiacich s mobilnými sieťami 4G/5G znáročňuje pochopenie všetkých procesov a úloh v rámci infraštruktúry. Simulačné nástroje a aplikácie však štúdium rôznych problematík značne zjednodušujú. Táto práca ponúka pohľad na históriu, súčasný stav a budúcnosť mobilných sietí 4G/5G podľa štandardu 3GPP, ktoré formujú spôsob, akým komunikujeme, pracujeme a žijeme v digitálnom veku dnes. Súčasne popisuje procesy návrhu aplikácií, ako už z pohľadu dizajnéra, tak aj z pohľadu vývoju samotnej aplikácie.

Táto práca cieľi na teoretický rozbor a porovnanie mobilných sietí 4G/5G, ktoré formujú spôsob, akým komunikujeme, pracujeme a žijeme v digitálnom veku dnes. Teoreticky rozoberie podrobným spôsobom, akým spôsobom siete 4G fungujú. Následne je vysvetlená vnútorná architektúra sietí s popisom jednotlivých entít, ktoré sú pre chod sietí potrebné.

Nasledujúcou časťou práce je vysvetlená nová generácia sietí 5G a jej široké možnosti využitia. V tejto kapitole sa práca venuje rovnakým aspektom ako pri predchádzajúcej verzii 4G. Podrobne je prebratá problematika nasadení sietí 5G a spolupráca s predošlou generáciou sietí. Záverom tejto kapitoly je podrobné porovnanie sietí 4G a 5G, a vysvetlenie výhod novšej generácie. Sú rozobraté najdôležitejšie technológie, ktoré pribudli s novou generáciou sietí.

Záverečnou kapitolou práce je návrh aplikácie s ohľadom na všetky aspekty, od grafického dizajnu až po výslednú aplikáciu. Oblasť grafického návrhu aplikácie je detailnejšie preskúmaná, pričom je venovaná pozornosť najmä estetike, užívateľskej prívetivosti a efektívnemu vizuálnemu prezentovaniu komunikačných procesov v mobilných sieťach 4G/5G. V rámci tejto kapitoly je vysvetlený správny postup pri návrhu grafického rozhrania aplikácii a výber vhodného programu na realizáciu. Grafický návrh pokrýva jednotlivé komponenty, navrhuté pre aplikáciu a je vysvetlená ich úloha a funkčnosť.

V rámci poslednej kapitoly o návrhu aplikácie sú pokryté aj aspekty vývoja aplikácie. V tejto časti práce je vysvetlený postup pri vývoji aplikácie, od výberu programovacieho jazyka a nástrojov až po samotný vývoj aplikácie a jej výslednú funkcionálnosť.

1 Mobilné siete

Mobilné siete, sú telekomunikačné systémy navrhnuté na bezdrôtový prenos dát a hlasu cez mobilné zariadenia, ako sú smartfóny, tablety a iné mobilné koncové zariadenia. V minulosti slúžili výhradne na prenos hlasu. Dnes sú však vďaka evolúcii a digitalizácii bežne využívané okrem prenosu hlasu aj na rôzne iné účely ako napríklad pripojenie k internetu, či prenos užívateľských dát, prenos dát rôznych senzorov a podobne [1].

1.1 História mobilných sietí

Historia mobilných sietí začala s príchodom prvej generácie (1G) v 80. rokoch, kedy sa objavili prvé mobilné telefóny, umožňujúce hlasovú komunikáciu. S následným vývojom (2G, 3G, 4G) sa siete postupne zdokonaľovali, pridávali nové funkcie a umožnili rýchlejší prenos dát [2].

- 1G - bola prvou komerčne dostupnou mobilnou sieťou, ktorá fungovala na analógovom princípe a umožňovala iba hlasovú komunikáciu.
- 2G - úspešný prechod na digitálnu technológiu Global System for Mobile Communications (GSM) umožnil väčší počet súbežných hovorov a používateľov v sieti, poskytoval vylepšenú kvalitu hovoru a zavedenie krátkych textových správ (SMS).
- 2.5G - General Packet Radio Service (GPRS) priniesla datový prenos, umožňujúci pripojenie k internetu cez mobilné zariadenie a prenos malého množstva dát.
- 2.75G - Enhanced Data Rates for GSM Evolution (EDGE) rozšírenie 2.5G, ktoré zabezpečilo ešte vyššie prenosové rýchlosti dát, predstavujúce prechodný krok k 3G.
- 3G - The Universal Mobile Telecommunications System (UMTS) predstavila výrazné zlepšenie prenosových rýchlostí, umožňujúc široké spektrum multimédií, vrátane videokonferencií a mobilného prístupu na internet.
- 4G - Long Term Evolution (LTE) umožnila ešte vyššie rýchlosti dát a nižšiu latenciu, čo umožnilo náročné aplikácie, ako sú streamovanie vysoko kvalitných videí a online hranie hier.
- 5G - vychádza z LTE a podporuje obrovský počet pripojených zariadení, čím otvára dvere pre nové technologické inovácie vrátane Internet of Things (IoT) a Augmented Reality (AR) [2].

1.2 Frekvenčné spektrum

Frekvenčné spektrum v mobilných sieťach predstavuje rozsah rádiových frekvencií používaných na prenos dát medzi mobilnými zariadeniami a telekomunikačnými infraštruktúrami. Delí sa na rôzne pásma s rôznymi vlastnosťami [3].

V rámci rádiovej komunikácie je frekvenčné spektrum (rádiové spektrum) jedným z kľúčových zdrojov. Jednotlivé rádiové technológie využívajú preddefinované časti tohto spektra pre svoju prevádzku. S cieľom oddeliť rôzne skupiny technológií (s ohľadom na charakter prenosu a fyzikálne javy) je frekvenčné spektrum rozdelené a alokované pre špecifický typ bezdrôtového prenosu alebo technológie (napr. 87,5 - 108 MHz pre rozhlasové služby, 174-230 MHz pre televízne vysielanie v ČR). Kvôli obmedzeným zdrojom rádiového spektra a zabezpečeniu spoľahlivého fungovania rádiových systémov a technológií je dôležité spravovať jeho rozdelenie a monitorovať dodržiavanie tohto rozdelenia. Túto úlohu zabezpečujú štátne a medzinárodné inštitúcie (pre ČR: Český telekomunikačný úrad - ČTU) [3, 4].

Licenčné spektrum

Licenčné spektrum je časť spektra pridelená nezávislým spoločnostiam na prevádzkovanie ich technológií. Jednotlivé spoločnosti platia licenčný poplatok za určité spektrum. Licencie je možné získať od štátnych a medzinárodných inštitúcií (napr. ČTU). Výhodou je výhradné právo vysielat na pridelenej frekvencii, čoho následkom je minimalizácia rušení v alokovanej oblasti spektra, vplyvom prevádzky ostatých zariadení. Nevýhodou je obmedzenie v rámci určitej geografickej oblasti, ktorú má spoločnosť pridelenú [3].

Bezlicenčné spektrum

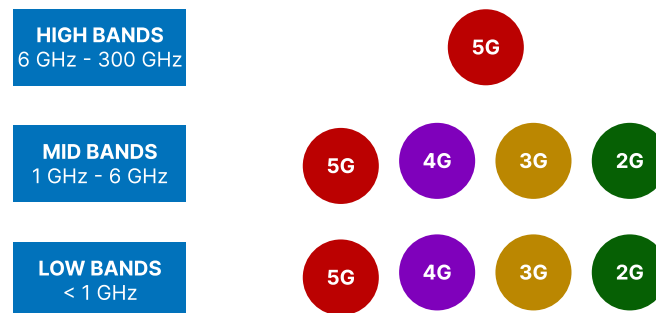
Bezlicenčné spektrum je časť elektromagnetického spektra, ktoré môže byť zdieľané s kýmkoľvek na nevýlučné použitie. Toto spektrum podlieha určitým regulačným obmedzeniam, ktoré musia byť dodržané. Vzhľadom k voľnému využitiu bezlicenčného spektra je nutné obmedzenie rádiových parametrov zariadení pre vzájomnú koexistenciu zariadení. Obmedzenie spočíva vo vysielacom výkone, veľkom počte zariadení - šum a bezpečnostné riziká, vzhľadom na neautorizovaný prístup k pásmu [3].

Pásmo ISM (Industrial, Scientific, and Medical) sú definované v rôznych frekvenčných pásmach a lišia sa v závislosti od regiónu a regulačných štandardov. Frekvenčné pásma často používané pre bezdrôtové komunikačné systémy, ako sú diaľkové ovládače, bezdrôtové senzory a domáce automatizačné zariadenia sú 433 MHz a 868 MHz. Frekvenčné pásmo 2,4 GHz a 5 GHz je široko využívané pre WiFi siete, Bluetooth. Technológia WiFi 6E (802.11ax), ktorá poskytuje ešte väčšiu šírku pásma a

menej interferencií vyžíva pásmo 6 GHz [5].

1.2.1 Rozdelenie spektra v rámci mobilných technológií

Frekvenčné pásmo (band) je rozsah frekvencií elektromagnetického vlnenia. V kontexte telekomunikácií a technológií sa tento termín často používa na popis frekvenčného rozsahu, v ktorých je daná mobilná technológia nasadená. V telekomunikáciách a bezdrôtových sieťach sa používajú rôzne frekvenčné pásma pre rôzne účely a aplikácie, rozdelené podľa frekvenčného spektra na: low-band, mid-band a high-band [6].



Obr. 1.1: Frekvenčné pásma - graf [6]

Low-band spektrum

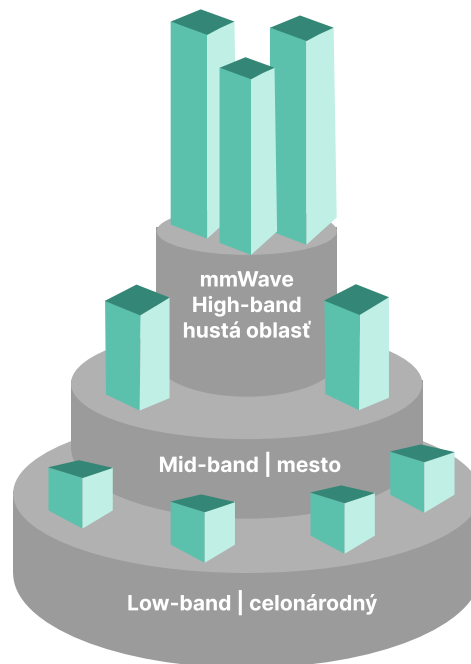
Nízkopásmové spektrum je akékoľvek spektrum nižšie ako 1 GHz na spektrálnom grafe. Prvé bezdrôtové siete, často nazývané analógové mobilné, boli zavedené v nízkopásmovom spektre 800 MHz. Nízkopásmové spektrum vo svete 5G umožňuje CSPs (Communications Service Providers) poskytovať komplexné (plošné) pokrytie vďaka fyzikálnej povahe vln [6].

Mid-band spektrum

Stredné pásmo spektra (1 GHz - 6 GHz) sa považuje za ideálne pre 5G, vďaka pomeru svojho pokrytia a rýchlosti. GSMA (Groupe Spéciale Mobile Association) opisuje spektrum v rozsahu od 3,3 GHz do 3,8 GHz ako ideálne, pretože mnohé krajiny na celom svete ho už určili pre 5G. Toto spektrum je kompromisom medzi low-band a high-band, pretože poskytuje relatívne slušné pokrytie pri vyšších rýchlostiach [6].

High-band spektrum

High-band spektrum je spektrum milimetrových vln (mmWave), ktoré je v rámci mobilných sietí definované ako pásmo od 6 GHz vyššie. Spektrum mmWave má zhoršené vlastnosti v oblasti šírenia signálu oproti mid-band a low-band spektrier a súčasne je náchylnejšie na okolité rušenie (napr. stromy, budovy a dokonca aj sklo) z dôvodu fyzikálnych vlastností šírenia vln. Výhodou high-band spektra je však možnosť alokovať výrazne viac prostriedkov pre rádiové technológie, čo výrazne zvyšuje prenosové rýchlosti [6].



Obr. 1.2: Frekvenčné pásma [6]

1.3 Štandardizácia mobilných technológií

S expanziou mobilných technológií na celom svete vznikla potreba vyriešiť otázku kompatibility medzi rôznymi technológiami a koncovými zariadeniami, a to nielen medzi krajinami, ale aj medzi rôznymi výrobcami. Na riešenie tohto problému sa v súčasnosti zameriava štandardizačná skupina The 3rd Generation Partnership Project (3GPP) [7].

1.3.1 3GPP

3GPP je partnerský projekt, spájajúci organizácie pre vývoj noriem v mobilných telekomunikáciách SDOs (Standards Development Organizations) z celého sveta. Projekt pôvodne slúžil na vývoj technických špecifikácií pre 3. generáciu mobilných, bunkových telekomunikácií UMTS (The Universal Mobile Telecommunications System) [7].

3GPP špecifikácie pokrývajú mobilné telekomunikačné technológie. Zahŕňajú technológie Radio Access Network (RAN), Backbone Network (Backbone) a služieb, ktoré poskytujú úplný popis systému mobilných telekomunikácií [7].

Čomu sa 3GPP venuje

Pôvodným predpokladom projektu 3GPP bolo vytvárať technické špecifikácie a technické správy pre mobilné 3G siete. Tieto siete boli založené na základných sieťach GSM a na technológiách rádiového prístupu, ktoré podporujú Universal Terrestrial Radio Access (UTRA) pracujúce s Frequency Division Duplex (FDD) aj Time Division Duplex (TDD) [7].

Rozsah bol následne upravený tak, aby zahrnul údržbu a vývoj technických špecifikácií a technických správ pre vyvinuté technológie 3GPP nad rámec 3G [7].

1.3.2 Spätná kompatibilita

Hlavným cieľom pre všetky 3GPP vydania je spätná aj dopredná kompatibilita systémov, všade, kde je to možné. Kompatibilita je dôležitá z dôvodu neprerušovanej prevádzky zariadení [7].

1.4 3GPP Release

3GPP (The 3rd Generation Partnership Project) používa systém paralelných "Vydání", ktoré poskytujú vývojárom stabilnú platformu pre implementáciu funkcií v danom bode a potom umožňujú prídanie nových funkcií v nasledujúcich vydaniach. Každý release predstavuje aktualizáciu štandardu technológie so zlepšeniami a novými funkciami. Technológia 1G nebola štandardizovaná 3GPP, pretože vznikla pred vznikom 3GPP. Bola to prvá komerčne dostupná mobilná telefónia, ale normy boli rôzne v jednotlivých krajinách. K dispozícii boli rôzne fázy a releasy, vrátane vydania 18, ktoré je aktuálne a vydania 19, ktoré je otvorené. [8]

- 2G GSM (Global System for Mobile Communications)
 - 2G Release 97
 - 2G Release 98

- 2.5G GPRS (General Packet Radio Service)
 - GPRS Release 97
- 2.75G EDGE (Enhanced Data rates for GSM Evolution)
 - EDGE Release 99
- 3G UMTS (Universal Mobile Telecommunications System)
 - 3G Release 99
 - 3G Release 4
 - 3G Release 5
- 4G LTE (Long-Term Evolution)
 - 4G Release 8
 - 4G Release 9
 - 4G Release 10 (LTE-Advanced)
 - 4G Release 11
 - 4G Release 12 (LTE-Advanced Pro)
- 5G
 - 5G Release 15
 - 5G Release 16 (známe ako 5G NR - New Radio)
 - 5G Release 17
 - 5G Release 18 - aktuálna
 - 5G Release 19 - otvorená [9]

Tab. 1.1: Tabuľka vydaní 5G [8]

Vydanie #	Stav	Funkčné zmrazenie	Dátum ukončenia
Vydanie 19	Otvorený	TBC (SA#109)	TBC (SA#110)
Vydanie 18	Otvorený	2023.12.15 (SA#102)	2024-06 (SA#104)
Vydanie 17	Zmrazený	2022.03.18 (SA#95)	2022.06.10 (SA#96)
Vydanie 16	Zmrazený	2020.07.03 (SA#88-e)	2020.07.03 (SA#88-e)
Vydanie 15	Zmrazený	2019.03.22 (SA#83)	2019.06.07 (SA#84)

2 Technológie 4G

Štvrtá Generácia predstavuje štandard pre mobilné telekomunikačné siete, ktorý priniesol výrazné vylepšenia oproti predchádzajúcim generáciám. 4G bolo vydané v decembri roku 2008, 3GPP Release 8. Celkovo vzaté, 4G predstavuje významný krok vpred v oblasti bezdrôtových komunikácií. Technológia bola navrhovaná s víziou dlhodobého udržovania a rozširovania LTE. 4G poskytuje spoľahlivé a vysokovýkonné pripojenie na mobilné zariadenia a umožňuje rôznorodé aplikácie pre moderný digitálny život [10].

2.1 Základné princípy 4G

Technológia 4G funguje s použitím inej sieťovej architektúry ako jej predchodcovia. Siete 4G používajú architektúru založenú na all-IP (internetovom protokole), čo znamená, že všetky údaje sa prenášajú pomocou rovnakého protokolu, aký používa internet. To umožňuje rýchlejší a efektívnejší prenos údajov, pretože všetky údaje sa prenášajú v paketoch [10].

Siete 4G tiež používajú techniku nazývanú Orthogonal Frequency Division Multiplexing (OFDM). OFDM rozdeľuje dostupnú šírku pásma na menšie subkanály, ktoré sa potom používajú na súčasný prenos údajov. To umožňuje efektívnejšie využitie dostupnej šírky pásma, čo vedie k väčšej kapacite siete a zvýšeniu rýchlosti prenosu dát [10].

Ďalším zlepšením sietí 4G oproti predchodcom je využitie technológie MIMO, ktorá využíva viacero antén na súčasné odosielanie a prijímanie údajov. Technológia MIMO je podrobnejšie spomenutá v kapitole 3.3.1 [10].

2.2 LTE

Long Term Evolution (LTE), je štandard (Release 8) pre bezdrôtovú širokopásmovú komunikáciu, ktorý je súčasťou 4G mobilných telekomunikačných sietí. LTE je prepracovaním štandardu 3G, aby uspokojil dopyt po prenose dát s nízkou latenciou [11].

Redizajn zahŕňa

- Kompletne založená na IP adresách
- Zjednodušená sieťová architektúra
- Nové rádiové rozhranie Evolved Universal Terrestrial Radio Access Network (E-UTRAN)

- Nová metóda modulácie OFDM
- Variabilná šírka pásma (1.4, 3, 5, 10, 15, 20 MHz) [11]

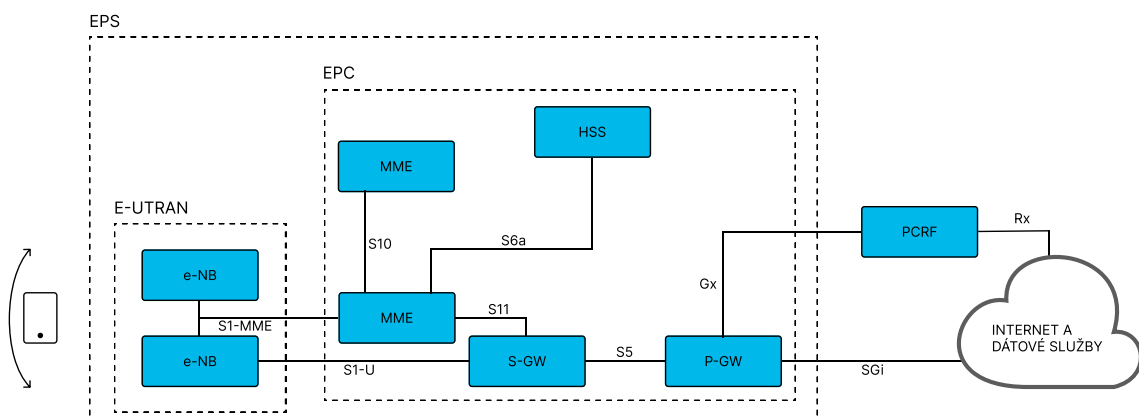
Bežne využívané frekvenčné rozsahy technológií LTE

- Band 1 (B1): 2100 MHz (1.9 GHz uplink, 2.1 GHz downlink)
- Band 3 (B3): 1800 MHz (1.7 GHz uplink, 1.8 GHz downlink)
- Band 7 (B7): 2600 MHz (2.5 GHz uplink, 2.6 GHz downlink)
- Band 8 (B8): 900 MHz (880 MHz uplink a 925 MHz downlink)
- Band 20 (B20): 800 MHz (832 MHz uplink a 791 MHz downlink) [12]

2.3 Architektúra 4G

Sieťová architektúra LTE, nazývaná Evolved Packet System (EPS) sa skladá z nasledujúcich dvoch kľúčových komponentov. Jednotlivé entity architektúry je možno vidieť aj na obrázku 2.1:

- Evolved Universal Terrestrial Radio Access Network (E-UTRAN)
- Evolved Pacet Core (EPC) [13]



Obr. 2.1: Architektúra 4G [13]

Užívateľské zariadenie

Vnútrotná architektúra užívateľských zariadení pre LTE je rovnaká ako u UMTS a GSM, čo sú Mobile Equipment (ME). Všetky komunikačné funkcie sú obstarávané pomocou Mobile Termination (MT). Dátové toky sú riadené pomocou Terminal Equipment (TE). SIM karta pre LTE zariadenia je známa ako Universal Integrated Circuit Card (UICC). Táto aplikácia je známa ako Universal Subscriber Identity Module (USIM). Identifikátory terminálu:

- IMEI - International Mobile Equipment Identity - unikátny identifikátor zariadenia - 15 znakov
- IMSI - International Mobile Subscriber Identity - unikátne číslo, uložené na SIM karte na identifikáciu užívateľa siete - 15 znakov
- ICCID - Integrated Circuit Card Identifier - celosvetové overenie SIM kariet - 19-20 znakov [13, 14]

E-UTRAN - Prístupová sieť

E-UTRAN predstavuje rádiový prístupový sieťový systém, ktorý zahŕňa základňové stanice e-NodeB (evolved NodeB). Tento systém je zodpovedný za bezdrôtový prenos dát medzi mobilnými zariadeniami a základňovými stanicami v sieti LTE. Základňová stanica môže pri pripojení k mobilnému zariadeniu LTE vykonávať dve hlavné funkcie:

- e-NodeB vysiela a prijíma rádiové signály obojsmerne medzi všetkými mobilnými zariadeniami pomocou funkcií analógového a digitálneho spracovania signálu rozhrania LTE.
- e-NodeB posiela príkazy všetkým svojim zariadeniam pre riadenie ich prevádzky a pripojenia [13].

EPC - Základná sieť

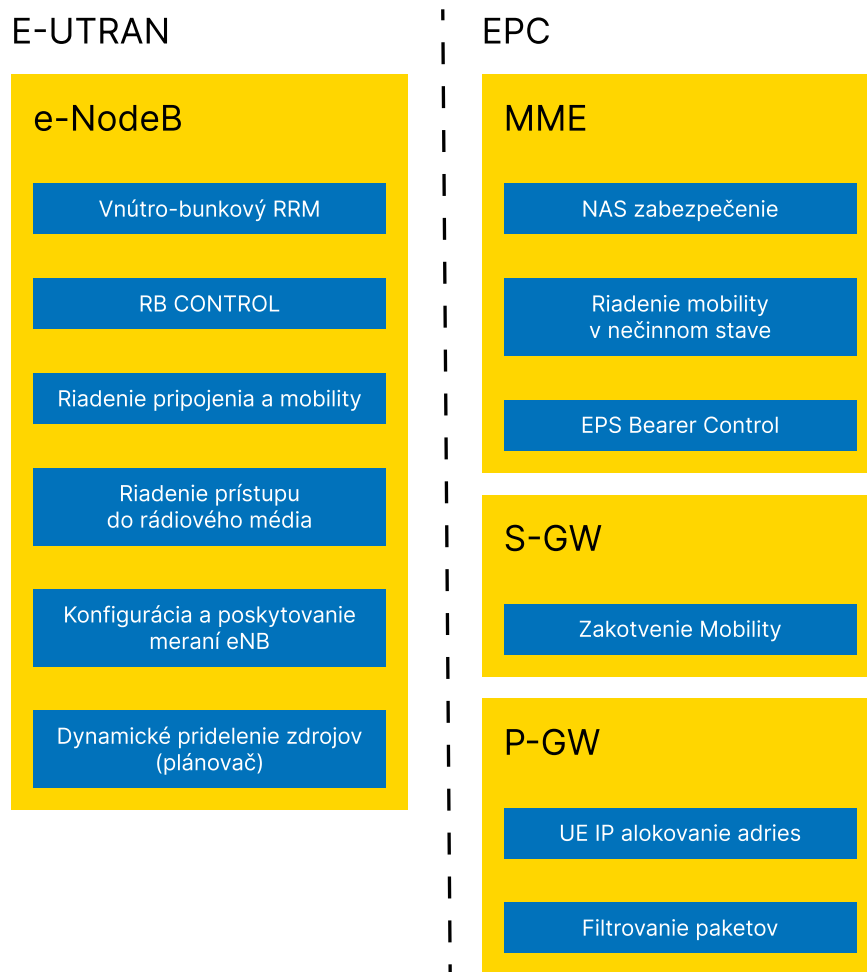
EPC (Evolved Packet Core) je chrbticová sieť systému LTE. Jeho úlohou je riadenie celej LTE siete. Medzi hlavné úlohy EPC patrí riadenie mobility, pripojenie do ďalších sietí a riadenie prístupu do siete. Architektúra EPC je zobrazená na obrázku 2.1, pričom určité časti z dôvodu jednoduchosti nie sú zobrazené, ako napríklad systém varovania pred zemetrasením a cunami - Earthquake and Tsunami Warning System (ETWS) a register identity zariadenia - Equipment Identity Register (EIR). Stručný popis jednotlivých komponentov a následne rozhraní EPC:

Prvky EPC

- HSS - Home Subscriber Server - je centrálna databáza užívateľov v sieti, ktorá obsahuje informácie o všetkých účastníkoch operátora siete (užívateľské profily, Quality of Service (QoS), blokovanie prístupu do siete a pod.).
- S-GW - Serving Gateway - funguje ako smerovač medzi základňovou stanicou a bránou PDN a spracováva presmerovanie údajov.
- P-GW - Packet Data Network Gateway - je kľúčovým komponentom EPC, ktorý zabezpečuje správu konektivity medzi sieťou LTE a externými paketovými dátovými sieťami, ako je internet alebo firemné siete (prideluje IP adresy, filtruje pakety a pod.).
- MME - Mobility Management Entity - je kritickým prvkom v jadrovej sietovej architektúre LTE, ktorý zabezpečuje efektívne riadenie mobility, pripojenia a autentifikácie pre mobilné zariadenia v sieti.
- PCRF - Policy and Charging Rules Function - umožňuje operátorom mobilných sietí implementovať a spravovať politiky a účtovacie pravidlá v sieti LTE, čo poskytuje flexibilitu a možnosť optimalizácie výkonu siete podľa konkrétnych požiadaviek a podmienok [13].

Rozhrania EPC

- S1-MME - rozhranie medzi eNodeB a MME, ktoré slúži na prenos signalizačných správ medzi bázovými stanicami a Mobility Management Entity a zabezpečuje riadenie mobility v sieti
- S1-U - rozhranie medzi eNodeB a S-GW, slúži na prenos dátových tokov medzi bázovými stanicami a Serving Gateway a zabezpečuje prenos dát medzi používateľskými zariadeniami a sieťou
- S5 - rozhranie medzi S-GW a P-GW, slúži na prenos dátových tokov medzi Serving Gateway a Packet Data Network Gateway a zabezpečuje smerovanie dátových tokov medzi 4G sieťou a externými sieťami
- S6a - rozhranie medzi MME a HSS, slúži na prenos informácií o používateľoch a ich službách medzi Mobility Management Entity a Home Subscriber Server a zabezpečuje autentifikáciu a autorizáciu používateľov
- Gx - rozhranie medzi PCRF a P-GW [13]



Obr. 2.2: Porovnanie úloh E-UTRAN a EPC [13]

3 Technológie 5G

Technológie 5G sú piatou generáciou mobilných sietí, významný vývoj dnešných sietí 4G LTE. Hlavným požiadavkom pri vývoji 5. generácie bol veľký nárast dát a konektivity dnešnej modernej spoločnosti. Nárast konektivity zahŕňal najmä predpoklad miliónov pripojených zariadení typu Internet of Things (IoT) na km², vyššie nároky na prenosovú rýchlosť a nižšiu latenciu, a „inovácie zajtrajška“. Technológie 5G zahŕňajú niekoľko technológií, ktoré sú navrhnuté podľa ich zamerania (využitia). Tieto technológie sú súčasťou ekosystému, kedy ich vzájomných kombinovaním je možné dosiahnuť unikátnych parametrov mobilnej siete. Tieto vertikály sú:

- URLLC - Ultra-Reliable Low-Latency Communication - kritické aplikácie, kde je prvoradá spoľahlivosť siete a nízka latencia
- eMBB - Enhanced Mobile Broadband - širokopásmové služby s vysokou rýchlosťou
- mMTC - Massive Machine-Type Communications - veľký počet zariadení pripojených do siete [15]

3.1 Mobilná sieť ako ekosystém

Využitie 5G sietí je široké a je možné rozdeliť ho do niekoľkých hlavných kategórií / vertikál:

mMTC

Massive Machine-Type Communications sa používa na pripojenie veľkého počtu zariadení a očakáva sa, že zmení priemysel IoT. Jeho cieľom je splniť požiadavky inteligentných miest a iných aplikácií IoT či Machine-to-Machine (M2M). Potenciál masívnej mMTC komunikácie je najmä v modernom priemysle a aplikáciách, poľnohospodárstve, výrobe a obchodnej komunikácii. Príkladom sú napríklad rôzne senzory, ktoré automaticky (bez ľudského zásahu) zaznamenávajú či odosielaajú dáta. Zahŕňa najmä technológie ako napríklad Narrowband-IoT a LTE Cat-M [16].

Hlavným rozdielom komunikácií IoT a M2M je interakcia s internetom. V základe, komunikácia Machine-to-Machine je rovnako strojová (napr. rôzne senzory), avšak, dáta ktoré sú zaznamenané nie sú odosielené do internetu ale spracované lokálne, iným zariadením. ITU-2020 stanovuje požiadavky pre masívnu strojovú komunikáciu vrátane nízkej spotreby energie, vysokého pripojenia zariadení, nízkej latency, vysokej spoľahlivosti a efektívneho využitia spektra [15].

eMBB

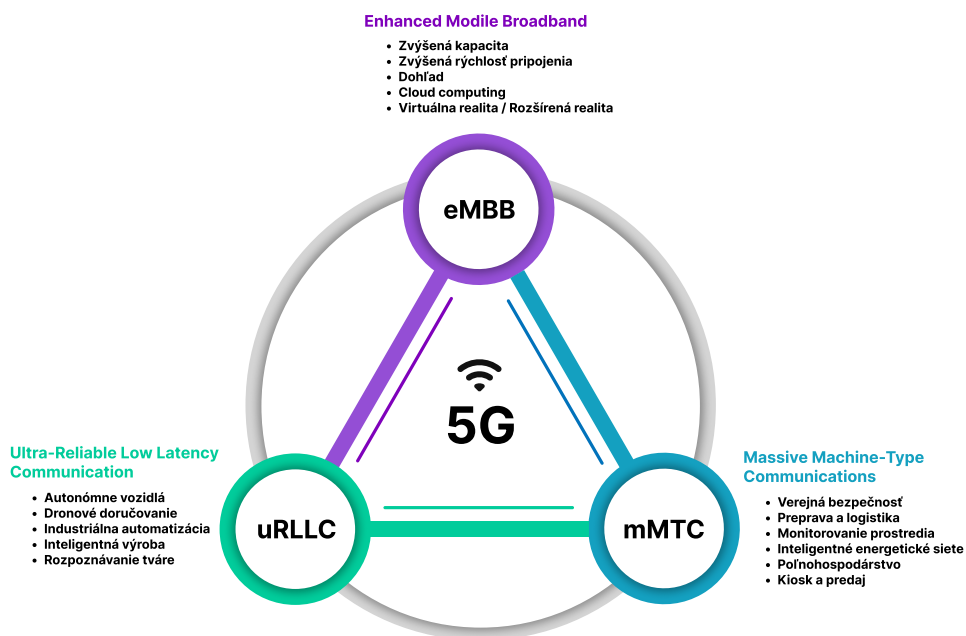
Enhanced Mobile Broadband je oblasť služieb ktorá je rozšírením súčasných mobilných dátových služieb pre širokú verejnosť. Hlavným bodom je poskytnutie vyššie rýchlosti sťahovania a nižšia latencia než v prípade sietí 4G. Táto kategória zahŕňa napríklad technológiu 5G New Radio (5G NR) [16].

Poskytuje rýchlosti až 10 Gb/s pre využitie vysokej šírky pásma, ako je streamovanie HD a 4K videa alebo hry Virtual Reality (VR) / Augmented Reality (AR). Nové aplikácie zahŕňajú stabilný bezdrôtový prístup na internet pre domácnosti, aplikácie vonkajšieho vysielania bez potreby vysielacích dodávok a lepšiu konektivitu pre ľudí na cestách [16].

URLLC

Ultra-Reliable Low-Latency Communication je ideálny pre kritické aplikácie, kde je prvoradá spoľahlivosť siete a nízka latencia (menej ako 1 ms) / riadenia v reálnom čase (real-time communication). Technológie spadajúce do tejto kategórie sú napr. Narrowband-IoT, LTE Cat-M, Reduced Capability (RedCap), 5G NR [16].

Do tejto kategórie spadá riadenie priemyselnej robotiky, komunikačných a bezpečnostných systémov, riadenie autonómnych a bezpečnostných systémov v doprave. Komunikácia s nízkou latenciou rovnako otvára možnosti vzdialenej lekárskej starostlivosti, procedúr a liečby [15].



Obr. 3.1: Možnosti využitia 5G [17]

3.2 Architektúra 5G

Architektúra 5G systémov je definovaná takým spôsobom, aby podporovala dátovú konektivitu a služby, ktoré umožnia nasadenie využívania techník ako Network Functions Virtualization (NFV) a Software Defined Networking (SDN) [18].

Kľúčové princípy a koncept

- Modularizovať funkčný dizajn - umožniť flexibilné a efektívne delenie siete.
- Umožniť každej Network Function (NF) priamu interakciu s externou NF ak je to potrebné. Architektúra nevyklučuje použitie medzilahlej funkcie na pomoc pri smerovaní správ riadiacej roviny (napr. Diameter Routing Agent (DRA)).
- Podpora jednotného autentifikačného rámca.
- Podpora „bezstavových“ NF, kde je výpočtový zdroj oddelený od úložného zdroja.
- Podpora súbežného prístupu k lokálnym a centralizovaným službám. Na podporu služieb s nízkou latenciou a prístupu k miestnym dátovým sieťom môžu byť funkcie UP nasadené v blízkosti prístupovej siete.
- Možnosť značnej virtualizácie jednotlivých prvkov siete Network Functions Virtualization (NFV)
- Možnosť pripojenia non-3GPP sietí pomocou non-3GPP Inter-Working Functions (N3IWF) rozhrania [18].

3.2.1 Standalone a Non-Standalone nasadenia

Z dôvodu jednoduchšieho prechodu a rýchlejšieho nasadenia novej generácie mobilných sietí 5G, bola v novom štandarde zahrnutá vzájomná kompatibilita medzi rádiovou sieťou 5G a hrubicovou sieťou 4G. Neskôr sa však postupne vyvíjali aj do samostatných sietí. Jednotlivé nasadenia, ktoré 5G ponúka sú Standalone (SA) a Non-Standalone (NSA), pričom, ako z názvu vypovedá, SA nasadenie funguje samostatne a NSA v spolupráci s už existujúcimi sieťami 4G LTE [19].

Non-Standalone

Technológia 5G NSA využíva 5G New Radio (5G NR) rádiovú prístupovú sieť na prenos užívateľských dát, pričom pre signalizáciu využíva existujúcu hrubicovú infraštruktúru 4G. Prenos signalizácie je realizovaný prostredníctvom tzv. kotvenia (anchoring), pri ktorom je zariadenie dodatočne pripojené k bunke technológie 4G, nazývanej kotva (anchor). 5G NSA využíva RAN technológie 5G, ktorá pracuje na

staršom jadre 4G LTE - známom ako Evolved Packet Core (EPC) - a riadi funkcie Control Plane (CP). Výhody NSA nasadenia:

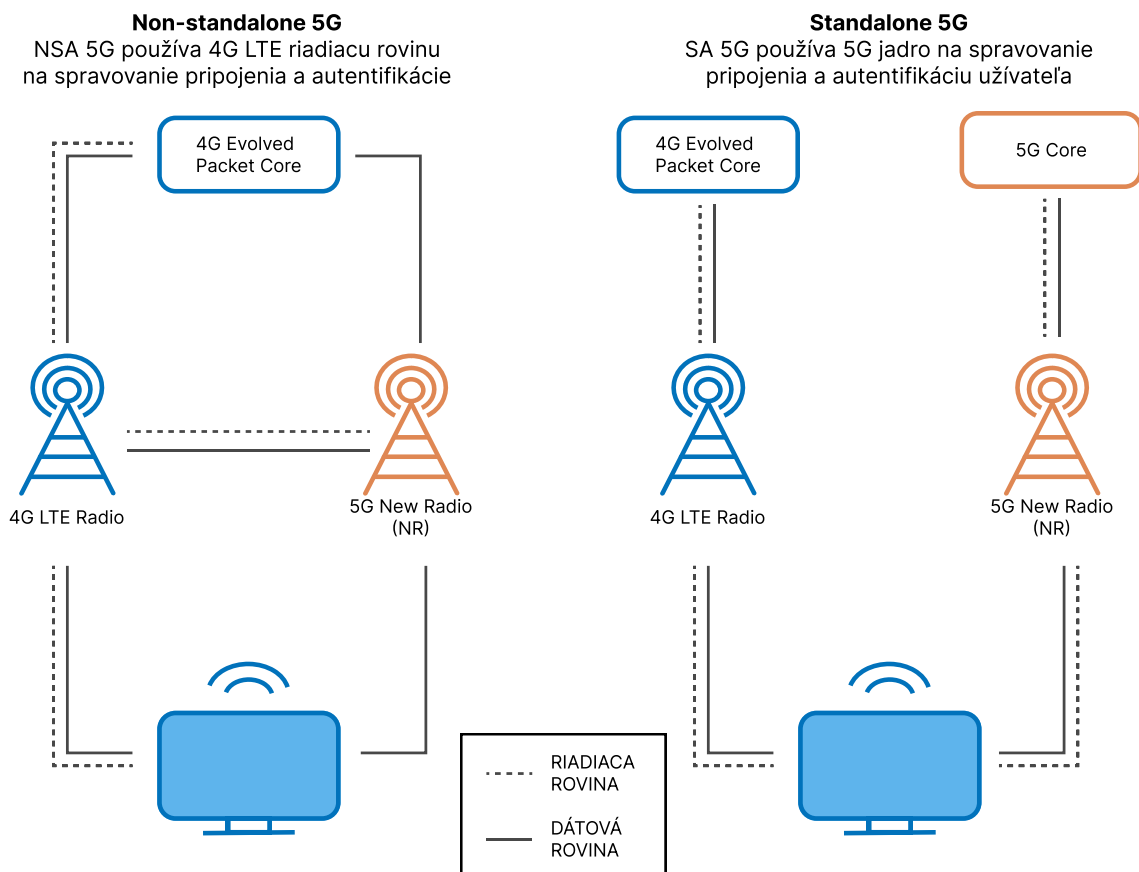
- Znížené náklady - vybudované na existujúcej infraštruktúre 4G
- Jednoduchšie a rýchlejšie nasadenie - využíva sa už existujúca Core infraštruktúra 4G [19]

Standalone

Siete 5G SA zahŕňajú 5G RAN aj cloudové natívne jadro 5G, čo sieťam NSA chýba a nahrádza sa jadrom 4G. Siete SA majú jadrá 5G, preto ponúkajú základné funkcie 5G a ich výhody. SA vyžaduje konfiguráciu novej architektúry zo strany MNO. Nevýhodou SA je nákladná a časovo náročná implementácia a správa spojená s preškolením personálu na novú infraštruktúru 5G. Výhody SA nasadenia:

- Znížená spotreba energie
- Širšie spektrum využitia ako napr. IoT a pod. [19]

Non-standalone 5G vs. standalone 5G



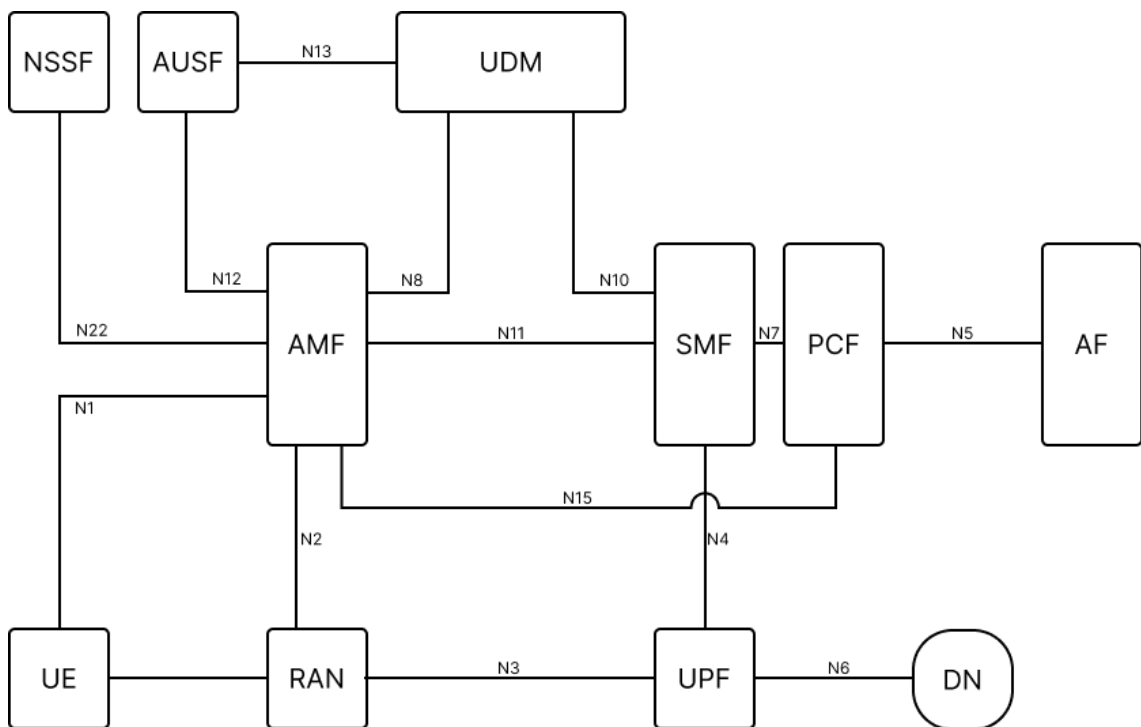
Obr. 3.2: Porovnanie Standalone a Non-Standalone [19]

3.2.2 Referenčný model architektúry

Architektúra 5G siete je definovaná ako „service-based“ a interakcia medzi sieťovými funkciami je reprezentovaná dvoma spôsobmi.

- Reprezentácia založená na service-based službách, kde sieťové funkcie v rámci CP umožňujú inéautorizované sieťové funkcie na prístup k iným službám. Táto reprezentácia taktiež zahŕňa point-to-point referencie bodov, kde je to nevyhnutné.
- Reprezentácia referenčného bodu ukazuje existenciu interakcie medzi službami NF v sieťových funkciách popísaných referenčným bodom point-to-point (napr. N11) medzi dvoma sieťovými funkciami (napr. AMF a SMF) [18].

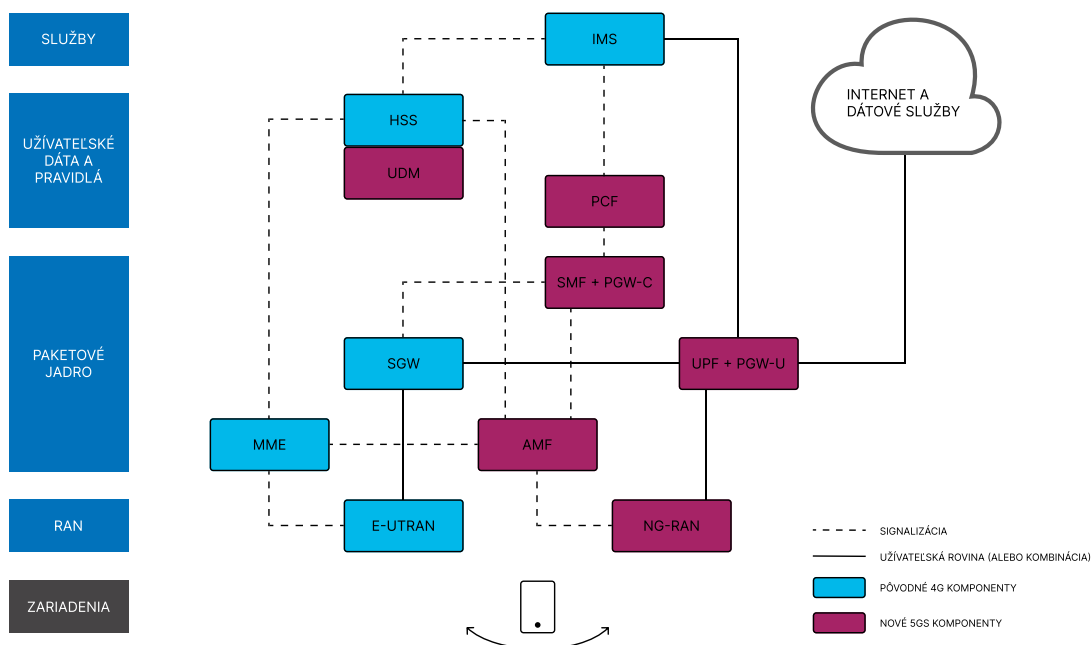
Obrázok 3.3 zobrazuje architektúru systému 5G v prípade non-roaming - s použitím reprezentácie referenčného bodu ukazuje, ako rôzne sieťové funkcie navzájom spolupracujú [18].



Obr. 3.3: 5G Non-roaming architektúra [18]

3.2.3 Model architektúry NSA

Obrázok 3.2.3 zobrazuje architektúru systému 5G v prípade NSA nasadenia. Architektúra obsahuje nové rozhrania a protokoly smerom k novej generácii RAN (NG-RAN) a zariadeniam, čo znamená, že migráciu RAN, vrátane aktív spektra a stratégií zariadení, je potrebné koordinovať so zavedením 5G [20].



Obr. 3.4: 5G NSA architektúra [20]

3.3 Porovnanie oproti staršej generácii - 4G

Siete 5G boli navrhnuté tak, aby pracovali v spojení so 4G sieťami. Využívajú na to množstvo makro buniek (macro cells), malých buniek (small cells) a vyhradených systémov v budovách (pico cells) [15].

Malé bunky sú typicky malé základňové stanice, dizajnované pre lokalizované použitie v malých vzdialenostiach, bežne od 10 metrov do jednotiek stoviek metrov. Tieto malé bunky poskytujú „výplň“ (in-fill) pre väčšie makro bunky. Bunky malých vzdialeností sú pre siete 5G nevyhnutnou súčasťou, pretože frekvencie mmWave majú veľmi malý krátky dosah pripojenia [15].

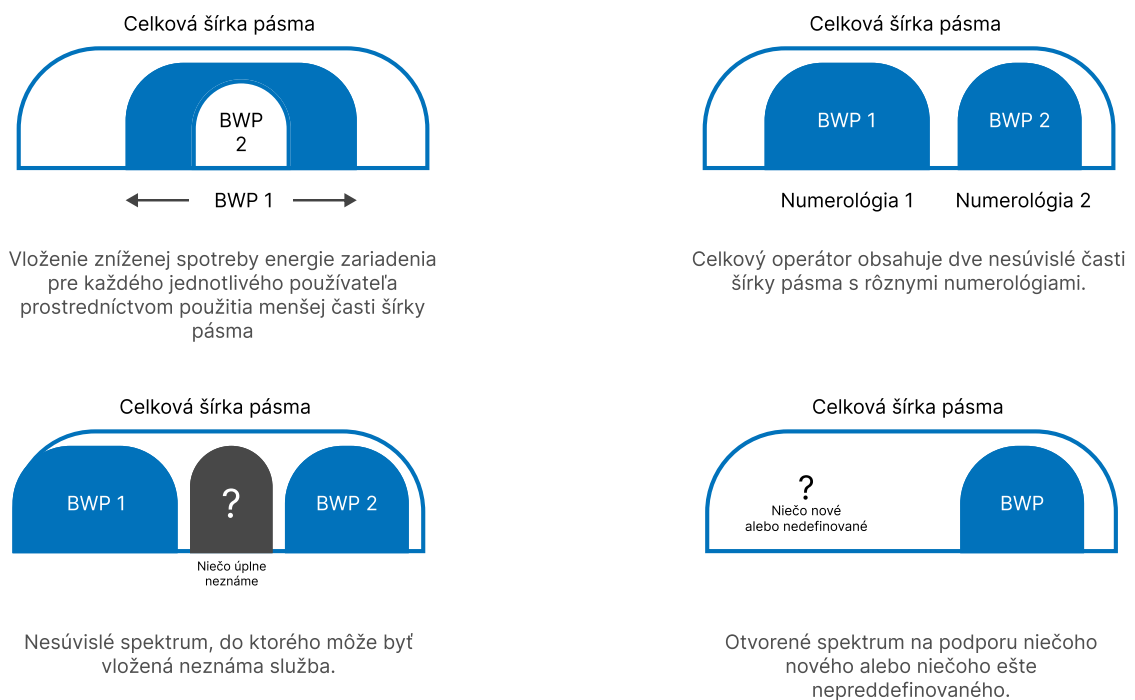
3.3.1 Zvýšená šírka pásma

V takmer všetkých krajinách sú počiatočné frekvencie pre 5G nižšie ako 6 GHz (v bežných prípadoch 3,3 - 3,8 GHz). Dodatočné mobilné spektrum nad 6 GHz,

vrátane pásiem 26 - 28 GHz, často označované ako mmWave, poskytne podstatne vyššiu kapacitu ako súčasné mobilné technológie [15].

Dodatočné spektrum a vyššia kapacita umožnia ako viac používateľov, tak aj viac dát a rýchlejšie pripojenie. Vzhľadom na postupný pokles používania starších sietí sa opätovne používajú nižšie pásma pre 5G a taktiež pre budúce prípady využitia [15].

Zvýšené spektrum v pásme mmWave (High-band) poskytuje lokalizované pokrytie, vzhľadom na nízky dosah signálu. Bežné šírky pásma (BW) pre pásmo mmWave sa pohybujú v rozsahoch do 400 MHz a pre pásmo mid-band nadobúda hodnoty 50 MHz, 100 MHz a až 200 MHz. Budúce nasadenia môžu využívať frekvencie v pásmach do 86 GHz [15].



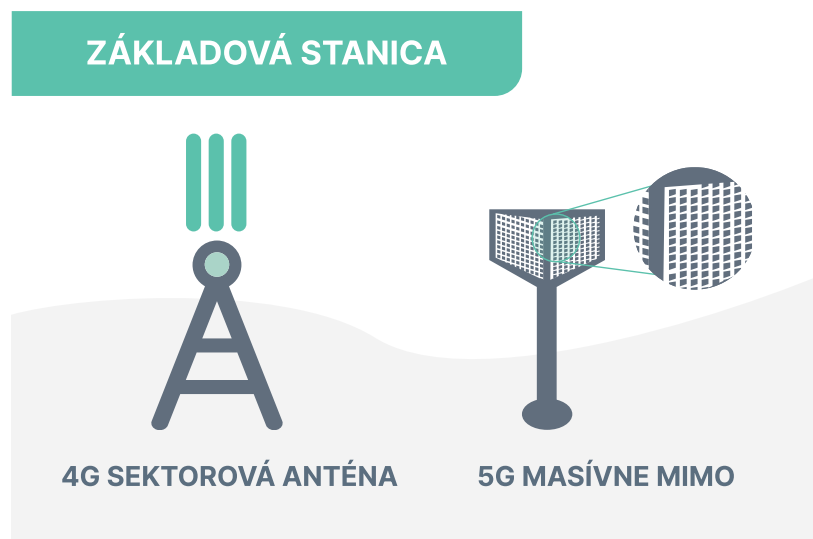
Obr. 3.5: Šírka pásma [21]

3.3.2 Masívne MIMO

Masívne MIMO (multiple input - multiple output) je kľúčová technológia v sieťach 5G. Jednoducho povedané, zahŕňa použitie veľkého počtu antén na základňovej stanici pre súčasné komunikovanie s viacerými užívateľskými zariadeniami. Stanica založená na viacnásobnom prístupe znamená - väčšiu kapacitu, viac užívateľov a rýchlejší prenos dát [15].

5G využíva masívne MIMO antény, ktoré obsahujú veľmi veľký počet anténnych elementov alebo pripojení na odosielanie a prijímanie väčšieho množstva dát súčasne. Výhodou pre užívateľov je pripojenie viacerých zariadení naraz a zachovanie vysokej priepustnosti [15].

Celková fyzická veľkosť masívnych MIMO antén 5G systémov je podobná ako u 4G antén. Ich vyššia frekvencia však umožňuje, že veľkosť jednotlivých elementov antény je menšia, čo umožňuje viac elementov (viac ako 100) v rovnakom fyzickom púzdre. Zariadenia 5G, vrátane mobilných telefónov obsahujú zabudovanú technológiu antény MIMO pre frekvencie mmWave [15].

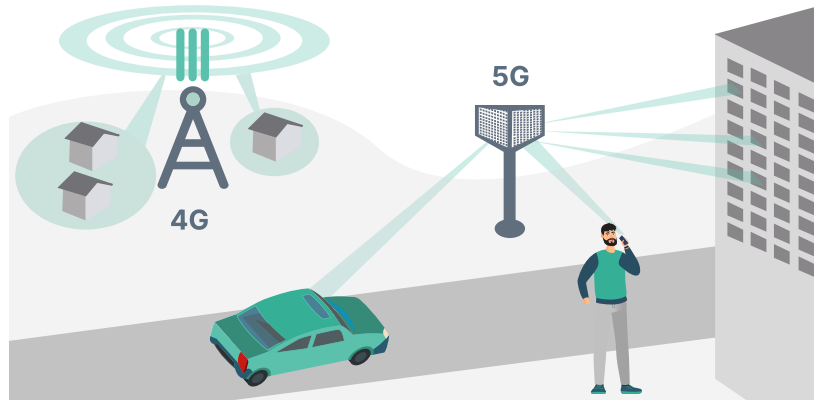


Obr. 3.6: MIMO anténa [15]

3.3.3 MIMO - Beam forming

Technológia riadenia lúčom (beam forming) umožňuje masívnym anténam základňových staníc MIMO smerovať rádiový signál k zariadeniam a nie všetkými smermi. Pre správne riadenie lúča sa využívajú pokročilé algoritmy spracovania signálu na určenie najvhodnejšej cesty, po ktorej sa rádiový signál dostane k zariadeniu. Táto technológia zvyšuje účinnosť a znižuje rušenie (nežiadúce rádiové signály) [15].

TECHNOLÓGIA RIADENIA LÚČOM



Obr. 3.7: MIMO Beam forming [15]

3.3.4 Nižšia latencia

Nižšia latencia znamená rýchlejšie odpovede zariadeniam, pripojeným v sieti. Pre zníženie latencie boli nevyhnutné úpravy v chrbticovej sieti (Core) aj v sieti rádiového prístupu (RAN). Nízka latencia a vysoká spoľahlivosť bezdrôtového rozhrania si vyžaduje nové techniky na minimalizáciu časových oneskorení v rámci niekoľkých TTI (Time Transmit Intervals). Spolu s robustnosťou a zlepšením kódovania na dosiahnutie vysokého stupňa spoľahlivosti (napr. jedna správa z miliardy je oneskorená alebo stratená) [15].

Tab. 3.1: Tabuľka latencií 4G a 5G

Technológia	Čas odpovedi
4G - LTE systems	20-30 ms
5G - eMBB	4-5 ms
5G - URLLC	1 ms

Zmeny v Core Network

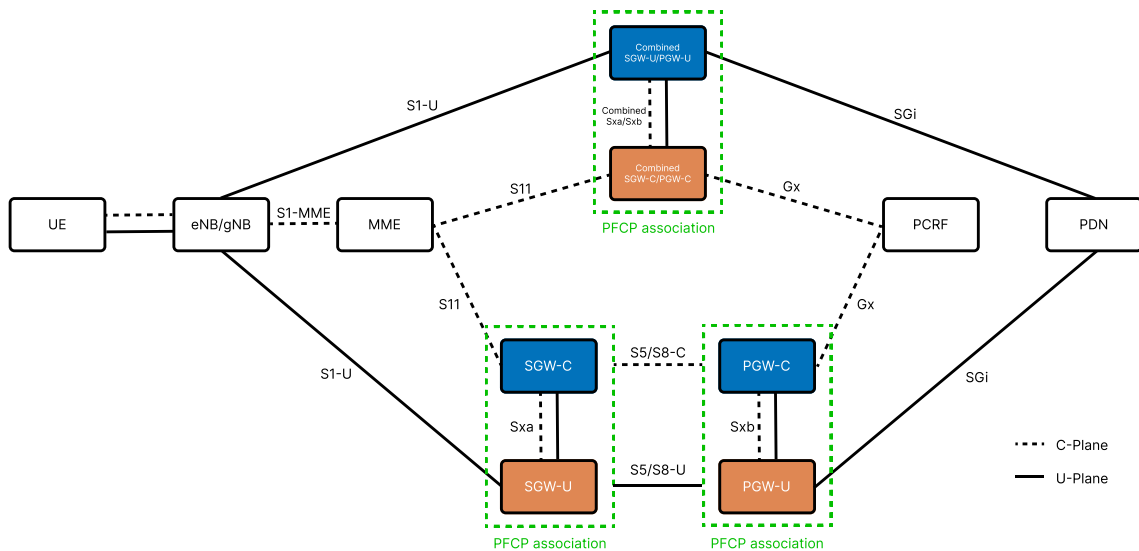
Vďaka prepracovanej chrbticovej sieti a distribuovaným serverom je kľúčovou vlastnosťou presunúť dáta čím bližšie ku koncovému používateľovi a skrátiť cestu medzi zariadeniami. Príkladom sú služby streamovania videa na požiadanie, kde je možné ukladať kópiu alebo „cache“ obľúbeného obsahu na lokálnych serveroch, takže čas pre prístup je menší [15].

Zmeny v Radio Access Network

Pre dosiahnutie nízkej latencie je potrebné prekonfigurovať RAN spôsobom, ktorý je vysoko flexibilný a softvérovo konfigurovateľný, aby podporoval veľmi odlišné charakteristiky typov služieb, ktoré systém 5G predpokladá [15].

Implementácia virtuálneho, dynamického a konfigurovateľného RAN umožňuje sieti fungovať s veľmi nízkou latenciou a vysokou priepustnosťou. Súčasne však umožňuje mobilnej sieti prispôbiť sa zmenám v sieťovom prenose, chybám v sieti a požiadavkám na novú topológiu [15].

V novej architektúre funguje takzvaný 4G/5G split RAN, kde užívateľská rovina (user plane - 5G) a riadiaca rovina (control plane - 4G) sú oddelené. To si vyžaduje oddelenie hardvéru na všeobecné použitie a hardvéru na špecializované použitie. Funkcionalita hardvéru na všeobecné použitie (nodes) je vhodná na virtualizáciu sieťových funkcií (NFV), kde sa špecializovaný hardvér v RAN stane dynamicky konfigurovateľný [15].



Obr. 3.8: Rozdelená užívateľská a riadiaca rovina [22]

4 Procesy v mobilných sieťach

V mobilných sieťach prebiehajú rôzne procesy, ktoré sú potrebné pre správne fungovanie siete. Rôzne procesy využívajú pri komunikácii rôzne protokoly a technológie. Práca sa zameriava na základné scenáre pripojenia a odpojenia zariadenia k sieti a využitie GTP protokolu.

4.1 GTP protokol

GRPS Tunnelling Protocol (GTP) je protokol, ktorý sa používa na tunelovanie dát v mobilných sieťach, založený na IP/UDP. Protokol je definovaný v štandardoch 3GPP a je používaný v sieťach GPRS, UMTS a LTE. Slúži na prenos dát medzi uzlami siete, ktoré sú v rôznych sieťových doménach.

GTP protokol môže byť rozdelený do dvoch hlavných častí: GTP-U a GTP-C, pričom GTP-U sa používa na prenos užívateľských dát a GTP-C na signalizáciu v Core sieti.

Hlavičky GTP umožňujú identifikáciu a spracovanie dát v rámci tunelového spojenia. Obsahujú informácie ako napríklad adresy zdroja a cieľa, typ správy, verziu protokolu a ďalšie metadáta potrebné na správnu komunikáciu medzi uzlami siete [23].

4.2 Pripojenie k sieti

Scenár prvého pripojenia k sieti (initial attach) je základný proces v mobilných sieťach 4G. Zahŕňa nadviazanie spojenia, autentifikáciu zariadenia, pridelenie IP adresy a pripojenie do siete. Scenár bol zahrnutý v simulácii a je možné ovplyvniť niekoľko faktorov, ktoré ovplyvňujú úspešnosť pripojenia zariadenia do siete.

RRC Connection Request: Zariadenie sa pokúsi nadviazať spojenie s bazovou stanicou. V prípade úspechu sa pokračuje ďalšími krokmi, v prípade neúspechu sa zariadenie pokúsi nadviazať spojenie znova. Využíva sa Random Access Channel (RACH) - kanál náhodného prístupu.

RRC Connection Setup: Po úspešnom nadviazaní spojenia sa zariadenie pokúsi získať prístup k sieti. Bazová stanica zodpovedá za pridelenie prístupu a nadviazanie spojenia.

RRC Connection Setup Complete: Potvrdenie o úspešnom nadviazaní spojenia. Správa obsahuje mobile identity - dáta o zariadení.

EMM Identity Request: Vyžiadanie si identity zariadenia. Správa obsahuje typ žiadanej identity zariadenia.

EMM Identity Response: Odpoveď na žiadosť o identity zariadenia. Obsahuje identifikátor zariadenia.

Authentication Request: Žiadosť od HSS pre zariadenie, ktoré sa pokúša pripojiť do siete. Nastane po vyžiadaní od MME, ktorá zistí, že zariadenie nie je autentifikované. Obsahuje náhodnú hodnotu RAND a autentifikačný vektor AUTN. Do MME je odoslaný XRES, vypočítaná hodnota, ktorá je pre autentifikáciu porovnaná s RES hodnotou zo zariadenia.

Authentication Response: Odpoveď na žiadosť o autentifikáciu odosielaná zo zariadenia do MME. Zariadenie porovná hodnotu AUTN a zhodnotí, či je sieť dôveryhodná. Obsahuje vypočítanú hodnotu RES, ktorá je vypočítaná z prijatej hodnoty RAND.

EMM Security Mode Command: Príkaz na nastavenie bezpečnostného módu. Správa obsahuje informácie o šifrovacích algoritmoch.

EMM Security Mode Complete: Potvrdenie o úspešnom nastavení bezpečnostného módu. Obsahuje IMEI zariadenia.

Create Session Request: Žiadosť o vytvorenie relácie. Správa obsahuje informácie o zariadení a žiadosť o pridelenie IP adresy.

Create Session Response: Odpoveď na žiadosť o vytvorenie relácie. Potvrdenie o úspešnom vytvorení relácie, obsahuje informácie o zariadení a sieti, ku ktorej sa pripája.

ESM Information Request: Žiadosť o poskytnutie dodatočných informácií o zariadení. Táto správa môže obsahovať požiadavky na údaje ako APN (Access Point Name), P-CSCF (Proxy-Call Session Control Function) adresu alebo ďalšie sieťové parametre.

ESM Information Response: Odpoveď na žiadosť o informácie o zariadení. Správa obsahuje informácie o zariadení a sieťové parametre.

UE Capability Enquiry: Správa slúži na vyžiadanie si povolení / práv daného zariadenia. Je odosielaná len v prípade, že informácie nie sú poskytnuté e-NodeB od MME. V scenári sa predpokladá prvé pripojenie a preto túto správu obsahuje.

UE Capability Information: Odpoveď s informáciami o povoleniach / právach zariadenia.

Security Mode Command: Odoslanie nastavenia bezpečnostného módu zariadeniu.

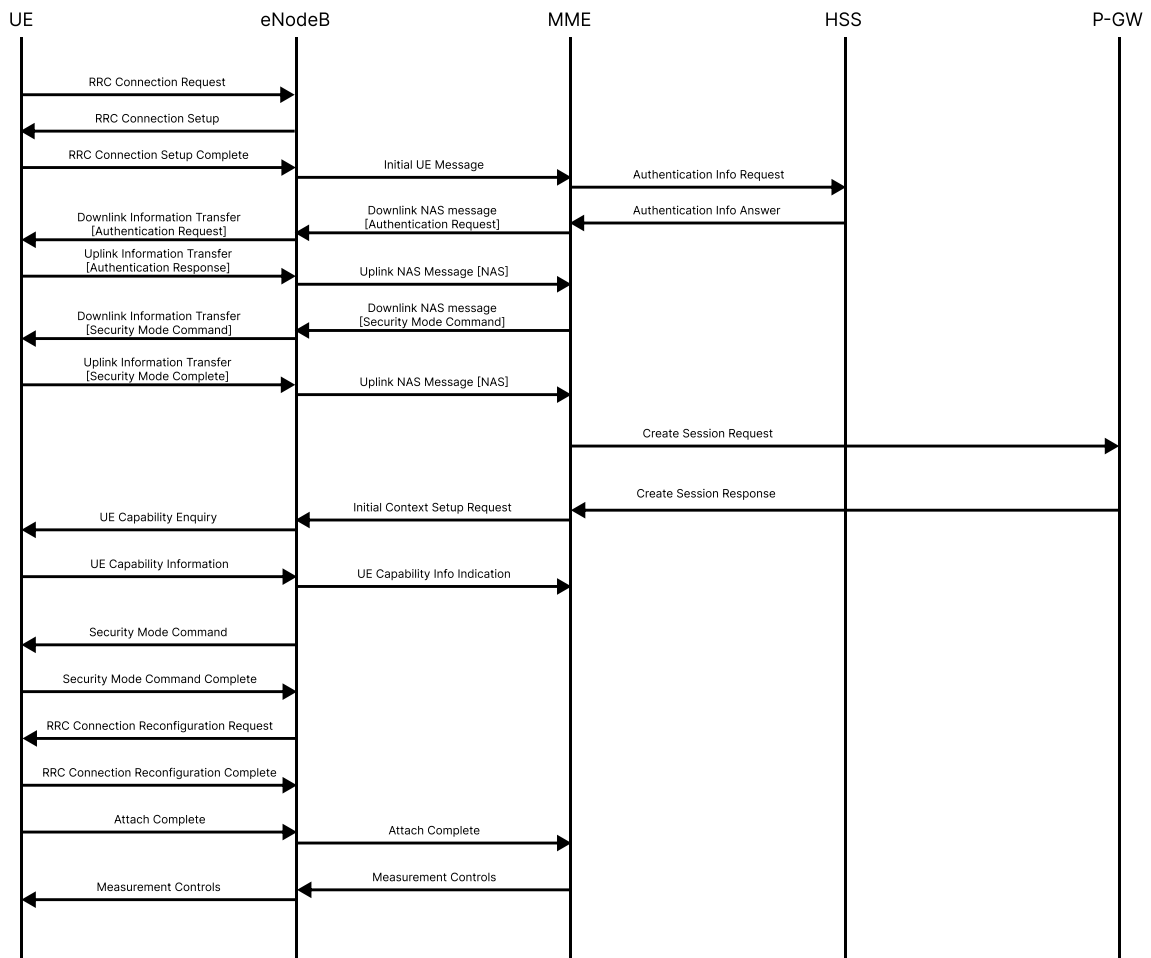
Security Mode Complete: Potvrdenie úspešnosti prijatia nastavení bezpečnostného módu.

RRC Connection Reconfiguration: Správa slúži na rekonfiguráciu spojenia. V prípade úspešného pripojenia je potrebné rekonfigurovať spojenie. Obsahuje kompletné informácie o RRC spojení.

RRC Connection Reconfiguration Complete: Potvrdenie o úspešnej rekonfigurácii spojenia.

Attach Complete: Potvrdenie o úspešnom pripojení zariadenia k sieti.

Measurement Controls: Správa slúži na kontrolu meraní. V prípade potreby môže byť zariadenie vyzvané na vykonanie meraní.



Obr. 4.1: Pripojenie k sieti - graf

4.3 Odpojenie od sieti

Scenár odpojenia od sieti bol v aplikácii zahrnutý ako základný proces. Zariadenie sa pokúsi odpojiť od siete vlastnou iniciatívou. Proces je zložený z niekoľkých krokov, ktoré sú potrebné pre správne odpojenie zariadenia od siete.

Detach Request: Žiadosť o odpojenie zariadenia od siete. Správa obsahuje dôvod odpojenia a identifikátor zariadenia.

Delete Session Request: Žiadosť o zrušenie aktívnej relácie. Prebieha medzi MME a SGW, pričom MME žiada o zrušenie aktívnej relácie.

Delete Session Response: Odpoveď na žiadosť o zrušenie aktívnej relácie. Obsahuje informáciu o potvrdení alebo zamietnutí žiadosti o zrušenie relácie.

Detach Accept: Potvrdenie o úspešnom odpojení zariadenia od siete.

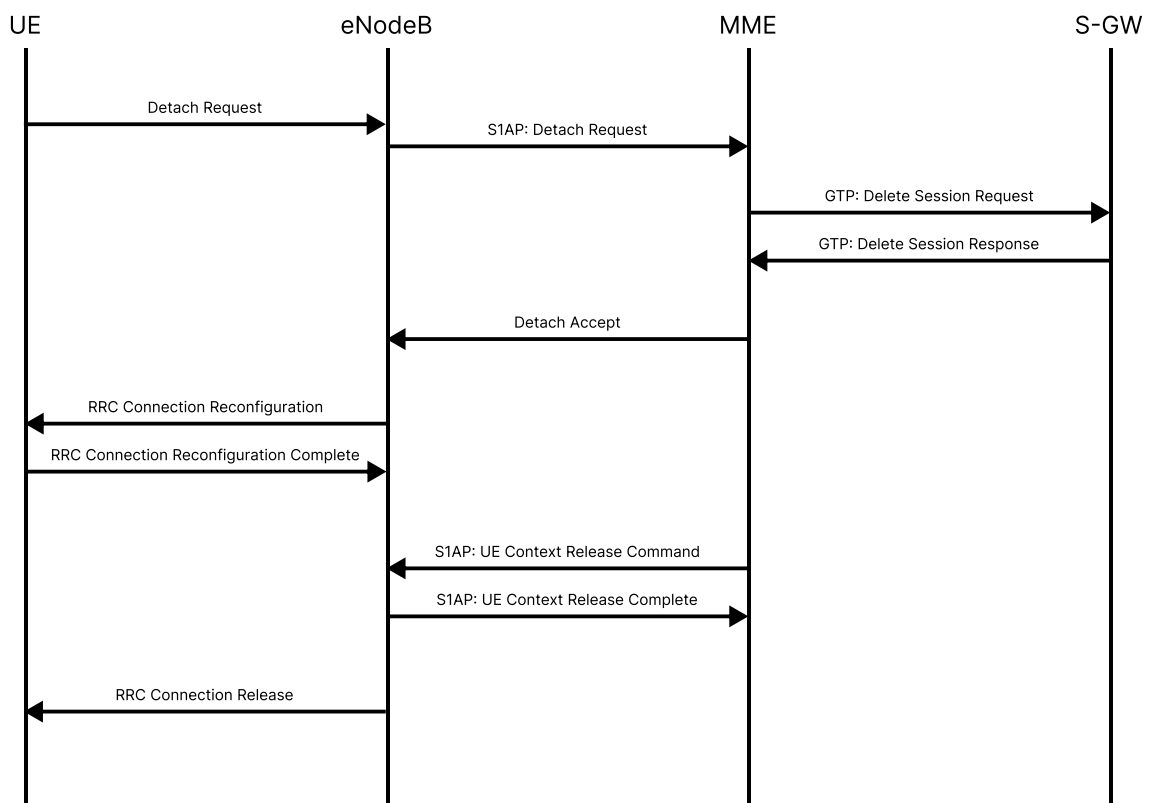
RRC Connection Reconfiguration: Správa slúži na rekonfiguráciu spojenia. V prípade odpojenia zariadenia od siete je potrebné rekonfigurovať spojenie.

RRC Connection Reconfiguration Complete: Potvrdenie o úspešnej rekonfigurácii spojenia.

UE Context Release Command: Príkaz na uvoľnenie kontextu zariadenia. Obsahuje identifikátory zariadení a dôvod uvoľnenia kontextu.

UE Context Release Complete: Potvrdenie o úspešnom uvoľnení kontextu zariadenia.

RRC Connection Release: Ukončenie spojenia medzi zariadením a sieťou. Správa obsahuje dôvod ukončenia spojenia.



Obr. 4.2: Odpojenie od siete - graf

5 Voľba nástrojov na realizáciu aplikácie

Určenie správnych nástrojov pre realizáciu aplikácie je kritickým krokom v procese vývoja softvéru. Voľba vhodných technológií a prostriedkov má zásadný vplyv ako na výkonnosť, škálovateľnosť a úspech celého projektu, tak aj na vývojársky komfort. Cieľom tejto kapitoly je identifikovať optimálne kombinácie nástrojov, ktoré zabezpečia robustnosť, flexibilitu a udržateľnosť aplikácie v dlhodobom horizonte.

Aplikácia vyvíjaná ako webová aplikácia, disponuje výhodou dostupnosti na všetkých zariadeniach s webovým prehliadačom. Preto je potrebné zvoliť vhodné nástroje pre vývoj webových aplikácií, ktoré zabezpečujú rýchlosť, škálovateľnosť a udržateľnosť aplikácie. Rovnako, je potrebné zvoliť nástroje pre vývojové prostredie, ktoré zabezpečujú rýchly a efektívny vývoj aplikácie. Záverečnou a súčasne kľúčovou časťou je zvolenie vhodných nástrojov pre beh aplikácie, či už pri vývoji alebo nasadení aplikácie.

Všetky tieto nástroje sú navzájom pri vývoji úzko prepojené a preto bolo potrebné zvoliť navzájom kompatibilné nástroje skôr, než bol vývoj aplikácie začatý. Väčšina nástrojov je konfigurovateľná pomocou určitého typu súboru, ktorý definuje konfiguráciu, balíčky, názvy, cesty, autorov a pod. (napr. `.Dockerfile`, `package.json`, `vite.config.js`, ...).

5.1 Nástroje potrebné pre beh aplikácie

Za nástroje pre beh aplikácie sa považuje napríklad server (aplikácia poskytujúca virtuálny server pre vývoj), kde je webová aplikácia dostupná pri vývoji aj pri nasadení. Ďalšou súčasťou, ktorú je nutné spomenúť je nástroj na kompiláciu (zostavenie) aplikácie, ktorý zabezpečí, že aplikácia je spustiteľná na serveri.

5.1.1 Kontajnerová virtualizácia

Kontajnerová virtualizácia spočíva v takzvaných kontajneroch, ktoré slúžia na izoláciu aplikácií a ich závislostí v oddelenom prostredí. V jednoduchosti, každý kontajner slúži obyčajne na beh jednej služby, pričom je možné spustiť viacero kontajnerov naraz a zabezpečiť komunikáciu medzi nimi.

Jedným z prostriedkov na kontajnerovú virtualizáciu je Docker - open-source platforma, ktorá umožňuje vývojárom vytvárať, spúšťať a spravovať kontajnery. Docker zabezpečí, že aplikácia je spustiteľná na akomkoľvek serveri či počítači, ktorý podporuje Docker kontajnery, čo je značnou výhodou pri nasadení aplikácií do produkcie. Výhodou aplikácie Docker je relatívne jednoduchá konfigurácia, rýchlosť spustenia aplikácie a možnosť použitia ako pri vývoji, tak aj pri nasadení.

Kontajnery je možné spustiť so základom, ktorý nie je potrebné konfigurovať (napr. Apache server, NodeJs základ a pod.). Pre bližšiu konfiguráciu kontajnerov sa využívajú súbory typu `.Dockerfile`, v ktorých je možnosť špecifikovať rôzne nastavenia kontajneru, kopírovať súbory či sťahovať ďalšie nástroje pre daný kontajner. Pri väčšom počte kontajnerov v projekte Docker umožňuje definovať konfiguračné súbory typu `.yml` so vzájomnými závislosťami (typicky `docker-compose.yml`). Ten je potom možné využívať aj na súčasné spustenie viacerých kontajnerov [24].

<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	5g-3gpp		Running (2/2)	0%		6 seconds ago	■ ⋮ 🗑️
<input type="checkbox"/>	app-1 b8884eb2	5g-3gpp-app	Running	0%	8080:80	6 seconds ago	■ ⋮ 🗑️
<input type="checkbox"/>	api-1 c00443ba1	5g-3gpp-api	Running	0%	8081:80	6 seconds ago	■ ⋮ 🗑️

Obr. 5.1: Docker kontajnery

5.1.2 Kompilátor

Kompilátor je nástroj určený na zostavenie zdrojového kódu do spustiteľného formátu. Jedným z moderných kompilačných nástrojov, primárne určených pre vývoj webových aplikácií je napríklad Vite. Jeho hlavným cieľom je poskytnúť rýchle a efektívne prostredie na vývoj moderných webových aplikácií pomocou nových technológií ako je napríklad Vue.js, React, alebo Svelte.

Jedným z hlavných rysov Vite je jeho použitie tzv. „esbuild“ kompilátora, ktorý je extrémne rýchly a dokáže efektívne spracovať moderné JavaScriptové a TypeScriptové kódy.

Hlavným konkurentom pre Vite je veľmi rozšírený webpack, ktorý je v súčasnosti jedným z najpoužívanejších kompilátorov pre vývoj webových aplikácií. V práci bol zvolený Vite z dôvodu jeho vyššej rýchlosti a efektívnosti oproti webpacku pri menších projektoch [25].

5.1.3 Správca balíčkov

Správca balíčkov pomáha s procesom inštalácie, aktualizácie, konfigurácie a odstránenia balíčkových závislostí. Yarn je uznávaný open-source správca balíčkov používaný na správu závislostí v JavaScriptových projektoch. Jeho hlavnou konkurenciou sú napríklad npm, pnpm a podobne, pričom yarn bol zvolený pre jeho možnosť prenosnej inštalácie konkrétnej verzie.

Yarn disponuje širokou škálou inovatívnych funkcií, ako sú workspaces, offline ukladanie do mezipamäte, paralelná inštalácia, interaktívne príkazy a ďalšie. Tieto vlastnosti zlepšujú produktivitu vývojára a zabezpečujú lepšiu orientáciu v projektovom adresári v projektoch menšieho aj väčšieho rozsahu [26].

5.2 Vývojové nástroje

Optimalizácia vývojového prostredia je kľúčovým krokom pre rýchly a efektívny vývoj aplikácie. Vývojové prostredie zahŕňa okrem IDE (Integrated Development Environment) aj rôzne nástroje, ktoré zabezpečia konzistenciu kódu, či jednoduchšie hľadanie chýb v kóde. Na to slúžia rôzne nástroje ako napríklad linter, formáter kódu, alebo nástroje na testovanie kódu.

5.2.1 Vývojové prostredie

Volba správneho vývojového prostredia je pri práci so strednými a väčšími projektami dôležitá. Visual Studio Code je vývojové prostredie vyvinuté spoločnosťou Microsoft, ktoré je optimalizované pre vývoj webových aplikácií. Toto vývojové prostredie je jednoduché na používanie, rýchle a flexibilné s možnosťou nainštalovania rozšírení (pluginov) pre široké spektrum jazykov či služieb.

5.2.2 Lintre a formáteri

Lint je nástroj, ktorý slúži na kontrolu kódu a hľadanie chýb v kóde. Hlavným rozdielom oproti formáteru je, že linter neupravuje kód, ale iba upozorňuje na chyby v kóde a prípadne navrhuje zjednodušenie/zmenu zápisu kódu. Formáter kód upravuje podľa prednastavených parametrov a zabezpečuje konzistenciu kódu (prevažne sa týka estetického hľadiska).

ESLint

ESLint je nástroj na identifikáciu a hlásenie vzorov nájdených v ECMAScript/JavaScript kóde, s cieľom zabezpečiť väčšiu konzistenciu kódu a predchádzať chybám. ESLint je plne rozšíriteľný, čo znamená, že každé pravidlo je zásuvným modulom a je možnosť pridávať ďalšie za behu. Taktiež môžete pridávať zásuvné moduly komunity, konfigurácie a parzery na rozšírenie funkcionality ESLintu [27].

Prettier

Prettier je nástroj, ktorý prináša zjednotenie do sveta formátovania kódu. Jeho najväčšou výhodou je ukončenie neustálych diskusií o štýloch a formátoch kódu.

Prettier je plne konfigurovateľný a je možné ho použiť s rôznymi jazykmi ako je napríklad JavaScript, TypeScript, CSS, a mnohé ďalšie. Použitie je jednoduché, po vytvorení konfiguračného súboru `.prettierrc.json` je možné spustiť príkaz alebo použiť nastavenie automatického formátovania po uložení súboru [28].

5.2.3 Git a GitHub

Git je distribuovaný systém na správu verziovania zdrojového kódu, ktorý umožňuje vývojárom sledovať zmeny v kóde, spolupracovať na projektoch a spravovať ich históriu. Umožňuje ukladať rôzne verzie kódu, vytvárať vetvy na experimentovanie s novými funkciami a potom ich zlúčiť späť do hlavnej vetvy [29].

GitHub je webová platforma postavená na Git, ktorá slúži na hosting a zdieľanie projektov so spoločenstvom. Poskytuje nástroje pre správu projektov, sledovanie problémov a diskusiu, čo z neho robí populárne miesto pre open-source spoluprácu a vývoj softvéru.

Po nahratí súborov na GitHub sa súbory uložia do "Git repozitára", čím sa začnú automaticky sledovať a spravovať ich zmeny [30].

5.3 Programovacie jazyky + Frameworky

Rozhodovanie o voľbe vhodných jazykov a frameworkov je kritickým prvkom pri vytváraní web aplikácií. Správna voľba týchto nástrojov môže mať významný vplyv na celkový vývoj, výkon a udržateľnosť projektu.

Rozhodovanie v tejto kapitole je možné rozdeliť do dvoch kategórií: frontend a backend aplikácie. Frontend aplikácie je časť zodpovedná za zobrazovanie a interakciu s používateľom, zatiaľ čo backend aplikácia zastupuje spracovanie dát a logiku aplikácie.

5.3.1 Frontend aplikácie

Frontend (Frontend) aplikácie je časť webovej aplikácie, ktorá interaguje priamo s používateľom. Zahrňuje všetky časti aplikácie, ktoré sú zobrazené a ovládané v prehliadači používateľa, ako sú užívateľské rozhrania, grafické prvky, formuláre a ďalšie interaktívne prvky.

Hlavným cieľom frontendu je poskytnúť používateľovi priateľské a interaktívne prostredie, v ktorom môže efektívne komunikovať s aplikáciou a vykonávať rôzne úlohy.

Pre realizáciu aplikácie boli zvolené nasledujúce jazyky a frameworky, ktorých spolupráca medzi sebou je vhodná pre vytvorenie frontend časti aplikácie. React je

framework pre tvorbu užívateľských rozhraní, ktorý je založený na jazyku JavaScript (TypeScript). Sass sa stará o štýly aplikácie a fonty s ikonami zabezpečujú zobrazenie ikon v aplikácii.

5.3.2 React + TypeScript

React

React je open-source JavaScript knižnica pre tvorbu užívateľských rozhraní. Vytvorila ju spoločnosť Facebook a používa sa na vývoj webových aplikácií s dynamickým obsahom. React umožňuje tvorbu komponentov, ktoré sa potom dajú znovupoužívať a komponovať do komplexných užívateľských rozhraní. Jeho hlavným princípom je jednosmerný tok dát, kde zmeny v stave aplikácie spôsobujú automatickú aktualizáciu užívateľského rozhrania. Každý komponent môže teda obsahovať svoje vlastné stavy a vlastnosti, ktoré sa dajú meniť a aktualizovať. React sa často kombinuje s ďalšími nástrojmi a knižnicami, ako napríklad React Router pre navigáciu v aplikácii [31].

TypeScript

TypeScript je nadstavba jazyka JavaScript, ktorá pridáva statické typovanie a ďalšie vylepšenia do jazyka. Umožňuje vývojárom definovať typy premenných, parametrov funkcií a návratových hodnôt, čím zvyšuje bezpečnosť kódu a uľahčuje jeho údržbu. Výsledný kód je kompilovaný do čistého JavaScriptu, čo umožňuje jeho spustenie vo všetkých moderných prehliadačoch a prostrediach Node.js [32].

Funkčné komponenty

Vytvorenie komponentu bežne spočíva vo vytvorení samostatného priečinku s názvom komponentu, ktorý obsahuje súbor s názvom `button.tsx`. V tomto súbore sa nachádza definícia komponentu, ktorý je vytvorený ako funkcia. Oddelenie komponentu do samostatného priečinka je práve kvôli prehľadnosti a jednoduchej identifikácii súborov, patriacich k nemu (napr. `types.ts`, `schemas.ts`).

Ukážka funkčného komponentu, ktorý zobrazuje tlačidlo s definovateľným názvom a href atribútom a jeho použitie:

Výpis 5.1: React komponent

```
import { FC } from 'react';

import { ButtonTypes } from './types';

const Button: FC<ButtonTypes> = ({ href, title }) => {
  return <a href={href}>{title}</a>;
};

export default Button;
```

Výpis 5.2: React typy

```
export type ButtonTypes = {
  title: string;
  href: string;
}
```

Výpis 5.3: Použitie komponentu

```
<Button href="https://www.google.com" title="Google" />
```

Hooky

Hook v Reacte je špeciálna funkcia, ktorá umožňuje React komponentom využívať stav (state), kontext a ďalšie funkcionality, ktoré by inak boli dostupné len v komponentoch. Niektoré základné hooky v Reacte sú useState na manipuláciu so stavom, useEffect na prácu s efektami ako načítavanie dát alebo odoberanie udalostí, a useContext na prácu s kontextom aplikácie. Používanie hookov umožňuje písať moderný, čistý a efektívny kód v Reacte [31].

Príklad použitia hooku na focus do input poľa pri nastavení focusOnMount atribútu na true:

Výpis 5.4: Použitie hooku

```
useEffect(() => {
  if (focusOnMount) {
    inputRef.current?.focus({ preventScroll: true });
  }
  // eslint-disable-next-line react-hooks/exhaustive-deps
}, []);
```

Kontext

React Context je funkcia, ktorá umožňuje zdieľať dáta medzi komponentmi v strome komponentov bez potreby prenášať dáta pomocou props. Je to užitočné v prípadoch, keď je potrebné preniesť rovnaké dáta cez viacero úrovní komponentov. Používa sa zvyčajne na ukladanie globálnych údajov ako napríklad stav používateľa, konfiguračné nastavenia alebo medzinárodné preklady.

V aplikácii je kontext použitý na ukladanie aktuálnych správ a detailu zvolenej správy. Tieto informácie je potrebné používať naprieč aplikácie a preto bol na uloženie zvolený kontext, v ktorom je obalená celá aplikácia. Pre použitie kontextu bol vytvorený komponent, ktorým je obalená celá aplikácia a následne je možné pristupovať k hodnotám kontextu pomocou hooku useApp. Príklad použitia kontextu na uloženie aktuálnej správy a detailu zvolenej správy:

Výpis 5.5: Vytvorenie kontextu

```
export const AppContext = createContext<{
  messages: Messages;
  message?: Message;
  detail?: AttachDetail | DetachDetail;
  commonSettings?: z.infer<typeof commonSettingsSchema> | null;
} | null>(null);
```

Výpis 5.6: Pužitie kontextu

```
return (
  <AppContext.Provider
    value={{
      messages,
      message,
      detail,
      commonSettings: commonSettings || defaultCommonSettings,
    }}
  >
    {children}
  </AppContext.Provider>
);
```

5.3.3 Štýly aplikácie

Sass

Sass je „stylesheet“ jazyk, ktorý je kompilovaný do Cascading Style Sheets (CSS). Umožňuje používať premenné, vnorené pravidlá, mixiny, funkcie a ďalšie, všetko

so syntaxou plne kompatibilnou s CSS. Sass pomáha udržiavať veľké štýly dobre organizované a uľahčuje zdieľanie dizajnu v rámci projektov a medzi nimi [33].

BEM

BEM (Block, Element, Modifier) je spôsob organizácie a štruktúrovania kódu pri vývoji webových stránok a aplikácií. Tento prístup kódovania je navrhnutý tak, aby bol systematický, modulárny a ľahko udržiavateľný. Snaží sa minimalizovať kaskádové štýly a konflikty štýlov pomocou jednoznačných identifikátorov pre každý blok, prvok a modifikátor. To zjednodušuje údržbu kódu a znižuje riziko nechceného ovplyvňovania štýlov medzi rôznymi časťami webovej stránky alebo aplikácie [34].

Výpis 5.7: Príklad použitia BEM

```
.form-item {
  &--error {
    .form-item__label {
      color: $c-error;
    }
  }

  &__help {
    font-weight: 400;
    font-size: $font-size-s;
    font-style: italic;
  }
}
```

Tailwind CSS

Tailwind CSS je moderný CSS framework, ktorý poskytuje sadu preddefinovaných tried a utility tried (nástrojových - trieda = CSS nastavenie) pre rýchle a efektívne vytváranie užívateľských rozhraní. Na rozdiel od tradičných CSS frameworkov, ktoré poskytujú štýlové komponenty, Tailwind sa sústreďuje na poskytnutie nízkoúrovňových utility tried, ktoré umožňujú vývojárom priamo definovať štýly priamo vo svojom HyperText Markup Language (HTML) kóde. Tieto triedy pokrývajú širokú škálu vlastností, ako sú farby, typografia, rozmiestnenie, padding, margin, a mnoho ďalších, ktoré je možné rozšíriť v konfiguračnom súbore o svoje vlastné možnosti [35].

Font s ikonami

Pre zobrazenie ikon bola využitá možnosť vytvoriť si vlastný font, ktorý obsahuje ikony. Tento font je možné použiť v aplikácii pomocou CSS tried, ktoré zabezpečia

zobrazenie ikony. Tieto triedy sú vytvorené pomocou Sass, ktorý nám dáva možnosti premenných a cyklov, ktoré následnegerujú triedy pre ikony. Pre vytvorenie samotného fontu je možnosť použiť akýkoľvek internetový generátor fontov, ktorý akceptuje ikony vo formáte .png/.svg.

Pre používanie boli vygenerované CSS triedy v 3 kategóriách: pozícia, veľkosť a samotná ikona. Triedu veľkosti nie je potrebné vždy definovať, pretože trieda pozície obsahuje automaticky aplikovanie veľkosti 16x16. To znamená, že napríklad pre aplikovanie ikony krížika na začiatok tlačidla je potrebné použiť triedy `'icon-before icon--cross icon--24x24'`.

5.3.4 Frameworky

Zod

Zod je knižnica pre validáciu dát v JavaScripte. Umožňuje definovať schémy a overovať, či dáta spĺňajú tieto schémy. Zod ponúka silné typovanie, flexibilitu a možnosti rozsiahleho nastavenia pre validáciu dát. Použitie Zod môže pomôcť zabezpečiť, že dáta spĺňajú očakávané formáty a štruktúry, čím sa minimalizuje riziko chýb a zvyšuje stabilita a spoľahlivosť kódu [36].

Výpis 5.8: Použitie zod

```
export const commonSettingsSchema = z.object({
  scenario: z.string(),
  progressFormat: z.string().default('counter'),
  timeStep: z.number().min(1).max(10),
});
```

Výpis 5.9: Použitie zod

```
export const messageSchema = z.object({
  id: z.string(),
  name: z.string(),
  position: z.string(),
  channel: z.string(),
  from: deviceSchema.nullish(),
  to: deviceSchema.nullish(),
  detail: z.union([attachDetailSchema, detachDetailSchema]),
});
```

Formik

Formik je knižnica pre React, ktorá uľahčuje správu formulárov v aplikáciách. Umožňuje jednoduchú a efektívnu manipuláciu so stavmi a spracovanie udalostí v reálnom čase v rámci formulárov. Formik poskytuje rôzne utility, funkcie a komponenty, ktoré pomáhajú s validáciou, manipuláciou s hodnotami, sledovaním stavu formulára a ďalšími súvisiacimi úlohami. Jej použitie je veľmi užitočné pri vytváraní komplexných formulárov v React aplikáciách, čo umožňuje zjednodušiť a zrýchliť proces vývoja [37].

Výpis 5.10: Použitie formiku

```
<Formik
  initialValues={{ firstName: '', lastName: '' }}
  onSubmit={async (values) => {
    await new Promise((resolve) => setTimeout(resolve, 500));
    alert(JSON.stringify(values, null, 2));
  }}
>
  <Form>
    <Field name="firstName" type="text" />
    <Field name="lastName" type="text" />

    <button type="submit">Submit</button>
  </Form>
</Formik>
```

Knižnica formik sama o sebe nie je plne kompatibilná s knižnicou zod, použitou na validáciu a preto bola aplikácia doplnená aj o balíček ‘zod-formik-adapter’, ktorý umožňuje jednoduchú integráciu medzi formikom a zodom. Rovnako pre zjednodušenie použitia komponentov formiku boli vytvorené vlastné komponenty (napr. FormikForm, FormItemInput, ...), ktoré sú založené na komponentoch formiku, ale sú prispôbené pre použitie v aplikácii. Validácia bola realizovaná pomocou zod schém. Pre odoslanie (submit) formulárov bol použitý komponent Button, ktorému bol nastavený atribút type na hodnotu submit. Jednotlivé vstupné hodnoty pre FormikForm boli definované v samostatných súboroch pomocou hookov hlavne kvôli prehľadnosti.

Výpis 5.11: Použitie FormikForm

```
<FormikForm {...formikProps} className="flex flex-col gap-4">
  <FormItemInput name="id" label="ID" />
  <FormItemInput name="name" label="Name" />
  <AllowedList />

  <SubmitButtons />
</FormikForm>
```

use-between

Knižnica use-between je menšia ale veľmi užitočná knižnica pre React, ktorá umožňuje zdieľať stavy medzi komponentami bez posielania pomocou props. Použitie je jednoduché, vytvorí sa hook, ktorý následne poskytuje zdieľané stavy a funkcie pre manipuláciu s nimi.

Výpis 5.12: Vytvorenie use-between

```
const useParams = () => {
  const state = useState<string>('paused');
  const lastMessage = useState<number>(0);
  const selectedDevice = useState<{ id: string; name: string }>();
  const selectedMessageId = useState<string>();

  return { state, lastMessage, selectedDevice, selectedMessageId };
};

export const useAppStore = () => useBetween(useParams);
```

Výpis 5.13: Použitie use-between

```
const {
  lastMessage: [lastMessage],
  selectedMessageId: [selectedMessageId, setSelectedMessageId],
} = useAppStore();
```

TanStack Query

TanStack Query, predtým známy ako React Query, je často opisovaný ako chýbajúca knižnica na získavanie údajov (data-fetching) pre webové aplikácie. Vo viac technických termínoch robí získavanie, ukladanie do vyrovnávacej pamäte, synchronizovanie a aktualizovanie serverového stavu vo webových aplikáciách jednoduchým a efektívnym. TanStack Query vyžaduje obalenie aplikácie alebo komponentu v `QueryClientProvider` a následne definovanie `QueryClient` s konfiguráciou [38].

Výpis 5.14: Konfigurácia TanStack Query

```
export const queryClientConfigDefaults = {
  defaultOptions: {
    queries: {
      retry: false,
      refetchOnWindowFocus: false,
    },
  },
} satisfies QueryClientConfig;
```

Výpis 5.15: Použitie TanStack Query Provider

```
const queryClient = new QueryClient(queryClientConfigDefaults);

const App: FC = () => {
  return (
    <QueryClientProvider client={queryClient}>
      <AlertProvider animationDuration={300} activeDuration={8000}>
        <AppProvider>
          <Toolbar />
          <Content />
          <Messages />
        </AppProvider>
      </AlertProvider>
    </QueryClientProvider>
  );
};
```

Použitie TanStack Query bolo realizované pomocou hooku `useQuery` a `useMutation`, ktorého vstupnými parametrami je niekoľko informácií ako `queryKey` / `mutationKey`, `queryFn` / `mutationFn` a ďalšie. Hodnota kľúča slúži na identifikovanie daného query, ktorý slúži napríklad na invalidovanie daného query a znovanačítanie dát. Funkcia `queryFn` / `mutationFn` je funkcia, ktorá sa spustí pri zavolaní hooku a získa dáta z Application Programming Interface (API). Hook následne vracia objekt, obsahujúci okrem dát, rôzne informácie potrebné pre chod aplikácie ako napr.: stav načítavania, chyby a pod.

Výpis 5.16: Použitie hooku useQuery

```
const {
  data: config,
  isLoading,
  isError,
} = useQuery({
  queryKey: [`app-settings_${selectedDevice?.id}`],
  queryFn: () =>
    getFromLocalStorage(
      `app-settings_${selectedDevice?.id}`,
      hssSettingsSchema
    ),
});
```

Výpis 5.17: Použitie hooku useMutation

```
const { mutateAsync } = useMutation({
  mutationFn: setToLocalStorage,
  onSuccess: () => {
    queryClient.invalidateQueries({
      queryKey: [`app-settings_${selectedDevice?.id}`],
    });
    newAlert(
      'Settings saved successfully',
      { alertType: AlertType.SUCCESS }
    );
  },
  onError: () => {
    console.error('Error saving object to local storage');
    newAlert('Error saving', { alertType: AlertType.ERROR });
  },
});
```

Výpis 5.18: Invalidovanie query

```
const queryClient = useQueryClient();

queryClient.invalidateQueries({
  queryKey: [`app-settings_${selectedDevice?.id}`],
});
```

React Flow

React Flow je knižnica pre tvorbu interaktívnych grafov a tokových diagramov v rámci webových aplikácií. Je postavený na základoch Reactu a umožňuje jednoducho vytvárať, upravovať a vizualizovať komplexné sieťové štruktúry. Často sa používa v oblastiach ako sú vizualizácie dát, plánovanie procesov, manažment úloh a mnohé ďalšie, kde je potrebné vizualizovať a manipulovať s komplexnými sieťovými štruktúrami [39].

Definícia prvkov (nodes) zahŕňa definíciu id, pozície (súradnice), data (rôzne vlastnosti) a type (typ prvku - react komponent). Ďalšou dôležitou časťou bolo definovanie prepojení (edges), pričom definícia spočívala v id, zdroji (source), ciele (target), prípadne type spojenia či konkrétnom bode prvku.

V práci bol React Flow použitý na vizualizáciu vnútornej štruktúry 4G siete a zobrazenie vzájomných vzťahov medzi jednotlivými komponentami. Niektoré funkcie ako napríklad zoom a pohyb poľom, boli zámerne zablokované, nakoľko manipulácia s jednotlivými prvkami (nodes) nie je žiadúca. Vizualizácia prvkov bola realizovaná pomocou vlastných komponentov, ktoré zahŕňajú modálne okno s nastaveniami, otvárajúce sa po kliknutí.

Výpis 5.19: React Flow Node definícia

```
{
  id: 'mme-1',
  position: { x: 800, y: 350 },
  data: { ...deviceTargets('mme-1') },
  type: 'MME',
},
```

Výpis 5.20: React Flow Edge definícia

```
{
  id: 'enb-mme',
  source: 'enb-2',
  target: 'mme-1',
  sourceHandle: 'enb-top',
  targetHandle: 'mme-left',
  type: ['mme-1_ue-1', 'ue-1_mme-1'].includes(message?.position || '')
    ? 'EdgeWithMessage'
    : 'step',
},
```

5.3.5 Backend aplikácie

Backendom aplikácie (Backend) je označovaná časť aplikácie, ktorá sa stará o spracovanie dát a logiku aplikácie. Zahrňuje všetky časti aplikácie, ktoré nie sú priamo zobrazené používateľovi, ale sú zodpovedné za spracovanie dát, komunikáciu s databázou a poskytovanie dát frontendu.

5.4 Grafické rozhranie

Tvorba úspešného webu alebo vývoj mobilnej či webovej aplikácie sú časovo i finančne náročné procesy, ktoré by sa nemali podceňovať. Návrh dizajnu aplikácií je možné rozdeliť do dvoch hlavných častí:

- UX dizajn - (User Experience) návrh užívateľskej skúsenosti
- UI dizajn - (User Interface) návrh užívateľského rozhrania [40]

Pri návrhu UI je ideálne postupovať systematicky. Prvými krokmi pre jednoduchý a súčasne efektívny návrh je napríklad predstava o dizajne, výber správnej farebnej kombinácie, či návrh jednotlivých komponentov, textov, ikoniek a podobne. Návrh jednotlivých komponentov, ikoniek pre font a vzhľad textov je možné vidieť v prílohách F.1 a G.1.

5.4.1 UX a UI dizajn

UX

Tento aspekt dizajnu sa zaoberá celkovým zážitkom, ktorý má užívateľ s aplikáciou. Zahŕňa porozumenie užívateľskej cesty, interakcií a emócií počas používania produktu. Kľúčovými aspektmi UX designu sú použiteľnosť, dostupnosť, informačná architektúra a celková spokojnosť užívateľa [40].

UI

UI design sa viac sústreďuje na vizuálne prvky a estetiku aplikácie. Zahŕňa vytváranie grafickej štruktúry, vizuálnych prvkov a celkového vzhľadu užívateľského rozhrania. Dizajnéri UI pracujú s aspektmi ako farebné schémy, typografia, ikony a usporiadanie interaktívnych prvkov, aby zabezpečili vizuálne príťažlivý a súdržný dizajn [40].

5.4.2 Programy na návrh UI dizajnu

Veľmi dôležitou časťou návrhu UI sú aplikácie. Je dôležité, aby zvolená aplikácia vyhovovala nie len samotnému dizajnérovi, ale aj osobe, ktoré bude návrh neskôr

používať pri vývoji aplikácie. Tento fakt je dôležitý, nakoľko, bežný vývoj aplikácie nie je len na jednej osobe, ale zvyčajne na celom tíme.

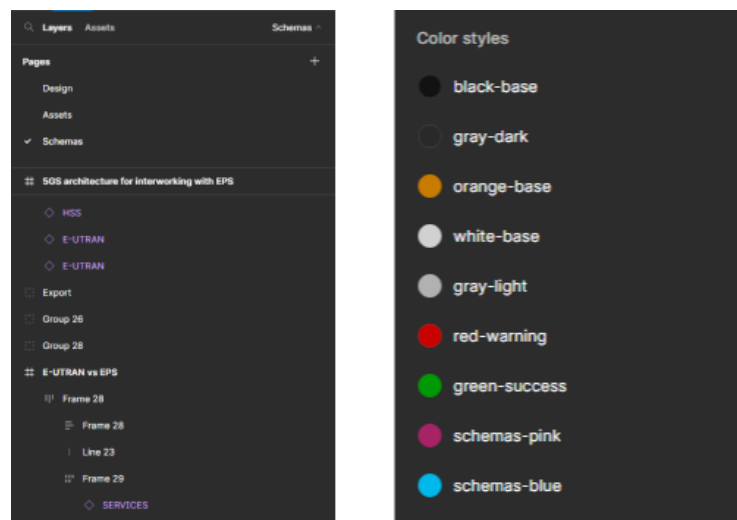
Pre návrh UI aplikácie bol v práci zvolený program Figma. Program bol zvolený z dôvodu jednoduchosti návrhu, možnosti následného preskúmania jednotlivých komponentov a v prípade nutnosti, možnosti generovania CSS štýlov pre web. Rovnako boli v programe navrhnuté schémy architektúr už spomínaných 4G/5G sietí.

Farebná paleta

Na obrázku 5.4.2 vpravo je možné vidieť ovládací prvok Figmy na definovanie farebnej palety, čo zahŕňal prvý krok návrhu UI. Vo farebnej palete boli definované 3 základné farby UI, doplnené o niekoľko odtieňov pre rôzne funkcie ako "hover" nad tlačidlami a o farby zobrazujúce úspešnú, či neúspešnú akciu pri stlačení tlačidla.

Vrstvy

Na obrázku 5.4.2 vľavo je možné vidieť ovládací prvok vrstiev. Práca s vrstvami obyčajne zahŕňa jednoduché pozicovanie, definovanie grid elementov, zoskupovanie či pomenovanie komponentov, ktoré môžu dediť informácie medzi sebou.



Obr. 5.2: Nástroje Figmy

6 Aplikácia pre simuláciu scenárov v mobilných sieťach

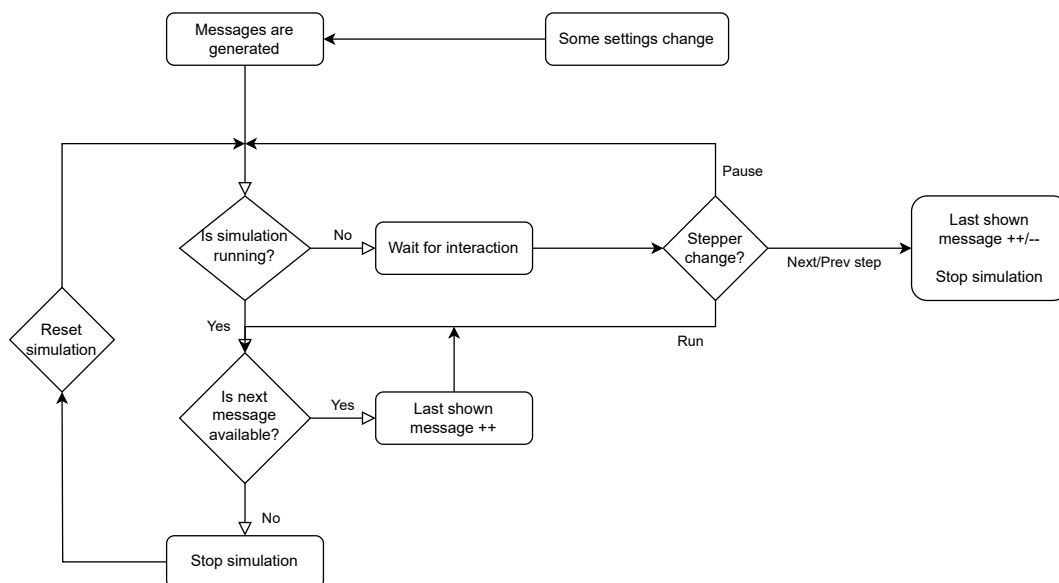
Cielom aplikácie bolo priblížiť a zjednodušiť používateľovi pochopenie procesov prebiehajúcich v mobilných sieťach. Aplikácia je určená na využitie v predmete MPC-KPM pre uľahčenie pochopenia komunikácie medzi jednotlivými entitami v mobilných sieťach 4G. Aplikácia je zameraná na simuláciu procesov v 4G sieťach, pričom cieľom nie je presné zobrazenie kompletnej komunikácie ale zdôraznenie najdôležitejších parametrov jednotlivých správ. V aplikácii je možné konfigurovať rôzne parametre scenárov, ktoré sú založené na reálnych procesoch v mobilných sieťach. Po spustení scenára je možné sledovať jednotlivé kroky scenára a zobraziť detaily o jednotlivých krokoch. Simuláciu je možné pozastaviť, krokovať, či upravovať parametre počas priebehu. Pri manuálnej zmene kroku je simulácia automaticky pozastavená a je nutné ju opäť spustiť v hlavnej lište.

6.1 Obecná štruktúra

Jednotlivé scenáre simulované v aplikácii, interaktívne reagujú na zmeny nastavení užívateľom. Zmeny v nastaveniach sú aplikované automaticky po uložení zmeny a okamžite prepočítané v hodnotách správ, ktoré daná zmena ovplyvní. Preto pri zmenách nie je nutnosť automaticky reštartovať simuláciu. Zjednodušený priebeh simulácie je zobrazený na obrázku 6.1.

Aplikáciu je možné rozšíriť o nové scenáre, prípadne o nové parametre jednotlivých správ. Štruktúra kódu a súborové zaradenie boli navrhnuté tak, aby bolo možné jednoducho pridávať nové scenáre a parametre aj bez znalosti celého kódu aplikácie.

Funkčne je aplikácia stavaná na princípe poľa správ, pričom každá správa musí obsahovať určité parametre. Povinnými a nepremennými parametrami sú id správy, názov správy, umiestnenie ikony správy v simulácii, zdroj a cieľ správy ako objekty a detail príslušnej správy. Parametre sú automaticky po každej súvisiacej zmene prepočítané a okamžite nahradené v kontexte aplikácie.



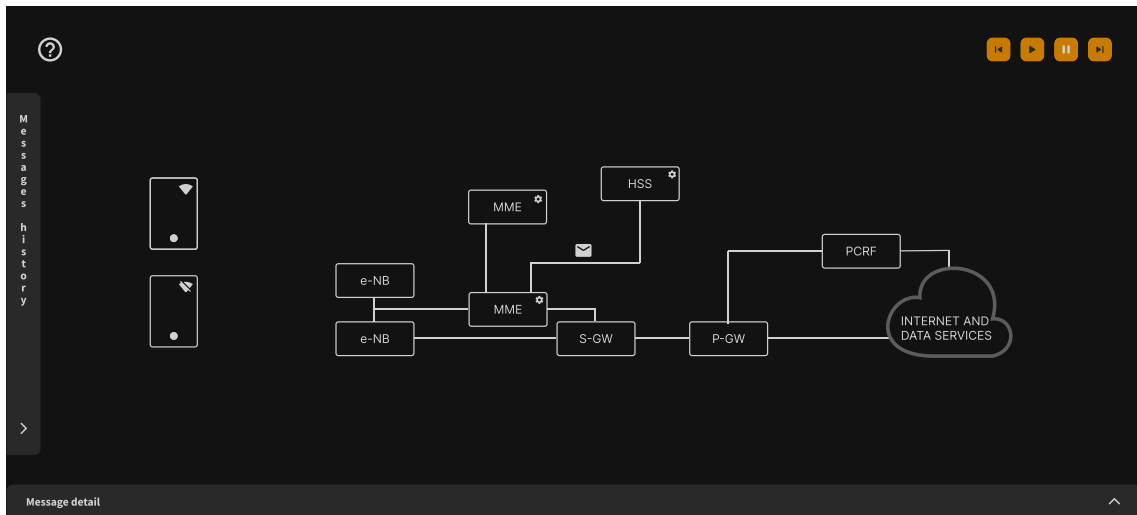
Obr. 6.1: Vývojový diagram aplikácie

6.2 Dizajn UI a funkcionality

Hlavné ovládacie komponenty boli umiestnené na hlavnú lištu aplikácie. Lišta zahŕňa globálne nastavenia aplikácie, tlačidlo pre resetovanie simulácie a v pravej časti krokovanie a pozastavenie simulácie.

Návrh dizajnu aplikácie

Vzhľadom na dnešné trendy v dizajne aplikácií, či už webových alebo mobilných, bol pre aplikáciu zvolený tmavý mód. Táto voľba nielen, že pridáva moderný a elegantný vzhľad, ale aj optimalizuje UX v nízkych svetelných podmienkach. Tmavý mód prispieva k zníženiu únavy očí a šetrí energiu zariadení. Celý návrh aplikácie je zobrazený v prílohe B.1. Vo výslednej aplikácii nastali drobné zmeny, ktoré boli vykonané z praktických dôvodov. Výsledný stav aplikácie je zobrazený v prílohe A.1.



Obr. 6.2: Grafický návrh aplikácie

Lišta nástrojov

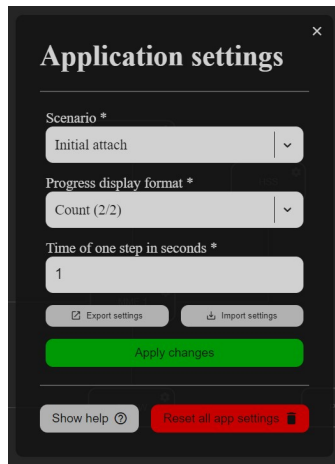
Ovládacie prvky pre riadenie simulácie a celkovo aplikácie boli umiestnené do praveho horného rohu. Boli použité tlačidlá s hlavnou farbou, práve z dôvodu UX, viditeľnosti a intuitívnosti.



Obr. 6.3: Ovládacie prvky aplikácie

Nastavenia aplikácie

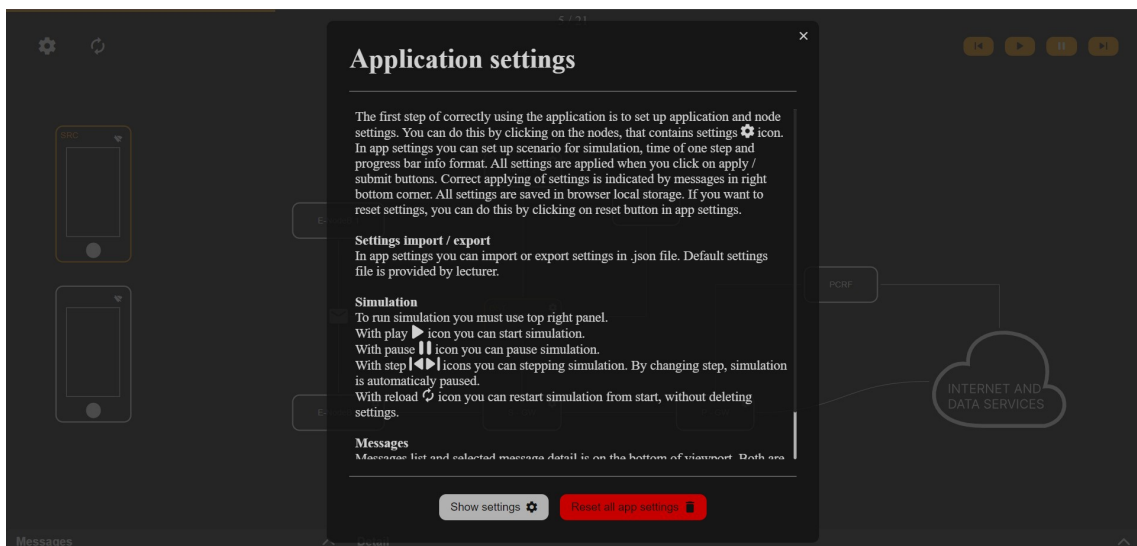
Nastavenia aplikácie obsahujú výber scenáru simulácie, nastavenie rýchlosti simulácie, nastavenie zobrazenia detailov o jednotlivých krokoch a nápovedu k aplikácii. Dôležitou časťou je import a export kompletnej konfigurácie simulácie, ktoré umožňuje užívateľovi uložiť a načítať prednastavené hodnoty.



Obr. 6.4: Nastavenia aplikácie

Nápoveda k aplikácii

Každá aplikácia, ktorá je komplexnejšia a nemusí byť z prvého pohľadu jasná úloha každého komponentu, by mala obsahovať nápovedu alebo návod. Z tohto dôvodu bolo v návrhu aplikácie umiestnené tlačidlo, ktoré by malo vzhľadom a polohou byť čo najviac prístupné a vypovedať o svojej úlohe. V pôvodnom grafickom návrhu bolo tlačidlo s ikonou otázniku pre nápovedu umiestnené priamo v ovládacej lište, neskôr bolo z praktického hľadiska presunuté do nastavení aplikácie. Jeho pozícia bola nahradená tlačidlom opakovania / resetovania simulácie. Po kliknutí na tlačidlo nápovedy v nastaveniach sa obsah modal okna nahradí nápovedou.



Obr. 6.5: Nápoveda aplikácie

Hlavný obsah

Hlavný obsah aplikácie obsahuje schému vnútornej architektúry 4G LTE sietí. Jednotlivé prvky 4G siete sú interaktívne prvky, zvýraznené ikonou nastavení v pravom hornom rohu (6.6), keď nastavenia obsahujú. Nastavenia prvkov sa otvárajú v modálnych oknách. Celkový výzor aplikácie je zobrazený v prílohe A.1.



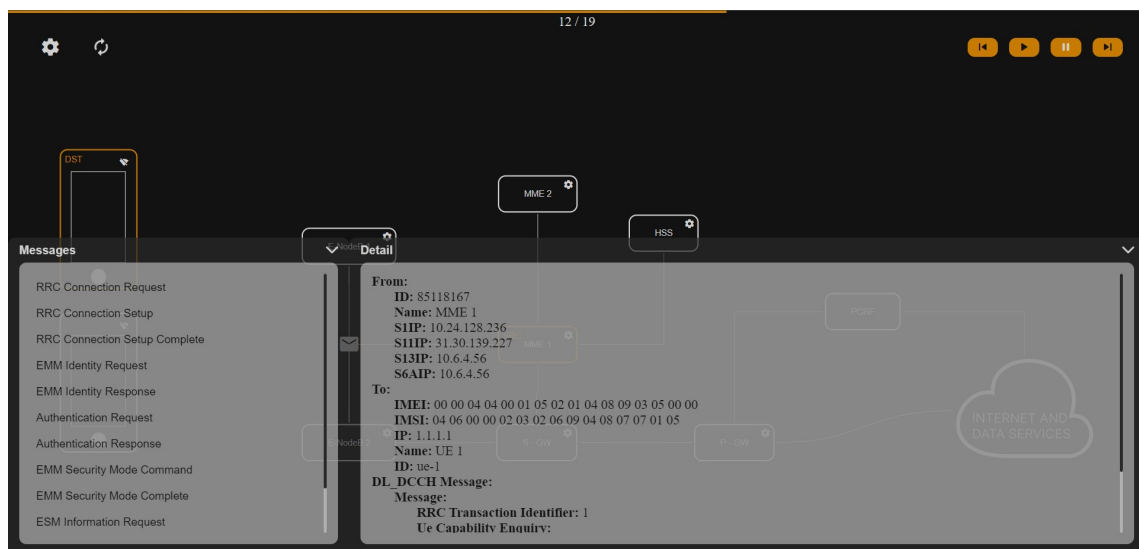
Obr. 6.6: Komponent architektúry v aplikácii

Lišta správ

Poslednými (vedľajšími) ovládacími prvkami aplikácie sú história a detail správ. V histórii je možnosť vidieť posledné odoslané správy s vhodným popisom (napr. zdroj - cieľ). Tento modul je možnosť sledovať pri behu simulácie a po kliknutí na správu sa zobrazí príslušný detail. Zvolená správa, ktorej detail je aktuálne zobrazený, je indikovaná ikonou lupy.

Detail správy je funkčne rovnaký rozbaľovač ako pri histórii, s rozdielom v zobrazovaných informáciách. Počas simulácie je automaticky zobrazovaný detail poslednej správy.

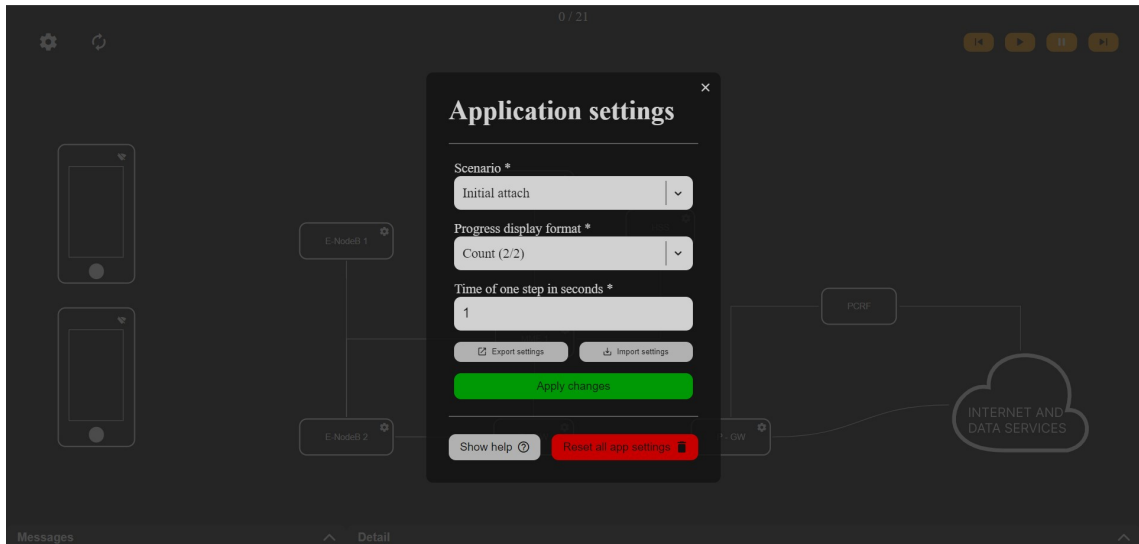
Návrh UI je možné vidieť v prílohe C.1 a D.1, realizáciu na obrázku 6.7.



Obr. 6.7: Detail správy

Modal okno

Modalové okno je všeobecný komponent, ktorý je používaný na rôznych miestach. Použitie modal okna je takmer neobmedzené, keďže obsah môže byť ľubovoľný. Hlavným princípom okna je zvýrazniť v popredí aplikácie určité dôležité informácie alebo nastavenia. Návrh UI je možné vidieť v prílohe E.1.



Obr. 6.8: Modal okno

Upozornenia

Po zmenách v nastaveniach aplikácie je vhodné používateľa informovať o úspešnom či neúspešnom vykonaní akcie. Preto bola aplikácia rozšírená o upozornenia, ktoré sa zobrazujú v pravom dolnom rohu aplikácie. Upozornenia je možné vytvoriť kdekoľvek v kóde aplikácie a sú globálne zobrazené postupne podľa času vytvorenia, každé na dobu 8 sekúnd.



Obr. 6.9: Upozornenia

6.3 Nastavenia prvkov

V simulácii je možné nastaveniami prvkov ovplyvniť priebeh jednotlivých správ. Každá zmena sa prejaví v správach automaticky po uložení daného nastavenia. Niektoré najdôležitejšie nastavenia aplikácie sú:

HSS

Obsahuje databázu používateľov, ktorí sú registrovaní v sieti. Túto zjednodušenú databázu je možné upravovať a pridávať či mazať záznamy IMSI povolených zariadení. Rovnako obsahuje HSS záznam o názvoch APN, ktoré sú porovnávané s APN nastavených v zariadeniach.

MME

Pri komunikácii sú dôležitými parametrami IP adresy, ktoré je pri MME možné nastavovať samostatne pre jednotlivé rozhrania.

Mobilné zariadenie

Pre autentizáciu zariadení sa využíva IMSI, ktoré je porovnané so záznamami v HSS. Z toho dôvodu obsahuje aj nastavenie zariadenia možnosť meniť IMSI a IMEI daného zariadenia. Ďalším ovplyvniteľným parametrom je pridelená IP adresa, ktorú v sieti zariadenie získa od P-GW. Pre zjednodušenie vysvetlenia priebehu je táto IP adresa ovplyvniteľná práve v nastaveniach zariadenia.

Záver

Táto práca bola zameraná na vývoj mobilných sietí od 4. generácie až po súčasnosť, ktoré sú kľúčovým prvkom dnešného digitálneho života. Koncept pripojenia, ktorý nie je viazaný na pevné miesto, nám umožňuje komunikovať, pracovať a žiť v pohybe, a to vďaka komplexným a vysoko výkonným systémom mobilných sietí 4G/5G. Dôraz bol kladený na teoretický rozbor sietí 4G a 5G vrátane popisu interných procesov a funkcionality aplikácie pre vizualizáciu a simuláciu 4G sietí.

V teoretickom zorbore bola popísaná funkcionality sietí 4G a 5G a následne prechod na siete 5. generácie, ktorý nám otvoril nové dimenzie možnosti využitia a spolupráce s predchádzajúcimi generáciami sietí. V práci bola popísaná interná stavba sietí, vrátane podrobného porovnania technologických výhod. Práca zahŕňa aj popis základných komunikačných procesov v mobilných sieťach umožňujúcich pripojenie a odpojenie zariadení k sieti.

Druhá časť práce sa zaoberala návrhom aplikácie, ktorá vizualizuje komunikačné procesy v mobilných sieťach 4G a 5G. V rámci tejto kapitoly boli pokryté všetky aspekty od grafického dizajnu až po vývoj aplikácie. Boli vysvetlené postupy pri návrhu grafického rozhrania a výber vhodných programov, jazykov a knižníc na jeho realizáciu. Pre realizáciu bol zvolený koncept webovej aplikácie, ktorá je dostupná na všetkých zariadeniach s webovým prehliadačom.

Aplikácia je postavená na javascript/typescript knižnici React, ktorá umožňuje jednoduchú modifikáciu a rozšírenie o ďalšie funkcionality v budúcnosti. Základné procesy v aplikácii sú realizované pomocou knižníc zodpovedajúcich na validáciu dát a vstupov, formik pre spracovanie formulárov a react flow pre zobrazenie štruktúry 4G siete. Na realizáciu dizajnu rozhrania aplikácie bola zvolená knižnica tailwind, poskytujúca sadu nástrojových tried a Sass - rozširujúci jazyk pre CSS.

Základným faktorom pri vývoji aplikácie bolo zabezpečiť jednoduchú a intuitívnu štruktúru kódu, ktorý bude možnosť jednoducho rozšíriť o ďalšie funkcionality a súčasne jednoducho nasadiť do produkcie. Vďaka prehľadnej štruktúre zdrojových kódov aplikácie, je možné rozšíriť aplikáciu o ďalšie funkcionality aj bez komplexnej znalosti celej aplikácie.

Výstupmi práce sú dve hlavné časti: teoretický rozbor mobilných sietí 4G/5G a aplikácia na vizualizáciu komunikačných procesov v týchto sieťach. Výsledná aplikácia sa skladá z 51 zdrojových súborov s obsahom najmenej 8500 riadkov formátovaného kódu. Simulácia poskytuje dva základné scenáre komunikácie v mobilných sieťach - pripojenie a odpojenie zariadenia, ktoré je možné simulovať v grafickom rozhraní aplikácie. Aplikácia poskytuje možnosť ovplyvniť výsledky simulácie zmenou parametrov a sledovať výsledky v reálnom čase.

Literatúra

- [1] DIALOGIC CORPORATION *Mobile Network (MN)* Online: v Brně, 2023. Dostupné z: <https://www.dialogic.com/glossary/mobile-network-mn>. [cit. 2023-12-05].
- [2] MPIRICAL *A Timeline from 1G to 5G: The Evolution of Mobile Communication* 2023-08-11 Online: v Brně, 2023. Dostupné z: <https://www.mpirical.com/blog/the-evolution-of-mobile-communication>. [cit. 2023-12-08].
- [3] TRENTON SYSTEMS *Licensed vs. Unlicensed Spectrum: Key Differences and 5G Use Cases* Nov 7, 2022 - Christopher Trick Online: v Brně, 2023. Dostupné z: <https://www.trentonsystems.com/blog/licensed-vs-unlicensed-spectrum>. [cit. 2023-12-03].
- [4] ČESKÝ TELEKOMUNIKAČNÍ ÚŘAD *O Úřadu* Online: v Brně, 2023. Dostupné z: <https://www.ctu.cz/o-uradu>. [cit. 2023-12-08].
- [5] MPIRICAL *Understand wireless short-range devices for global license-free systems* 2007-07-03 Online: v Brně, 2023. Dostupné z: <https://www.eetimes.com/understand-wireless-short-range-devices-for-global-license-free-systems/>. [cit. 2023-12-08].
- [6] NOKIA *5G spectrum bands explained — low, mid and high band* Online: v Brně, 2023. Dostupné z: <https://www.nokia.com/thought-leadership/articles/spectrum-bands-5g-world/>. [cit. 2023-12-04].
- [7] 3GPP A GLOBAL INITIATIVE *Introducing 3GPP* Online: v Brně, 2023. Dostupné z: <https://www.3gpp.org/about-us/introducing-3gpp>. [cit. 2023-11-08].
- [8] 3GPP *Správy* Online: v Brně, 2023. Dostupné z: <https://www.3gpp.org/specifications-technologies/releases>. [cit. 2023-12-05].
- [9] 3GPP *Release 17* Online: v Brně, 2023. Dostupné z: <https://www.3gpp.org/specifications-technologies/releases/release-17>. [cit. 2023-12-05].
- [10] VARSHNEY, Upkar. *4G Wireless Networks* Online: v Brně, 2023. Dostupné z: <https://ieeexplore.ieee.org/document/6235944>. [cit. 2023-12-07].
- [11] ETSI *4. generation (LTE)* Online: v Brně, 2023. Dostupné z: <https://www.etsi.org/technologies/mobile/4g>. [cit. 2023-12-07].

- [12] CLOVE TECHNOLOGY *LTE Frequency Bands and supported territories* Online: v Brně, 2023. Dostupné z: <https://www.clove.co.uk/pages/lte-frequency-bands-and-supported-territories>. [cit. 2023-12-10].
- [13] INTERVIEWBIT *LTE Architecture – Detailed Explanation* Online: v Brně, 2023. Dostupné z: <https://www.interviewbit.com/blog/lte-architecture/>. [cit. 2023-12-07].
- [14] EFANI *What is the Difference Between IMEI, IMSI, ICCID, and MSISDN Numbers?* 2023-05-15 Online: v Brně, 2023. Dostupné z: <https://www.efani.com/blog/difference-between-imei-imsi-iccid-and-msisdn-numbers>. [cit. 2023-12-10].
- [15] EMF EXPLAINED *5G EXPLAINED - HOW 5G WORKS* Online: v Brně, 2023. Dostupné z: <https://www.emfexplained.info/?ID=25916#What%20will%205G%20enable>. [cit. 2023-11-11].
- [16] ANTENOVA *What is mMTC in 5G? How does it work?* Online: v Brně, 2023. Dostupné z: <https://blog.antenova.com/what-is-mmtc-in-5g-how-does-it-work>. [cit. 2023-12-06].
- [17] C&T SOLUTION *What is 5G?* Online: v Brně, 2023. Dostupné z: <https://www.candtsolution.com/solutions/5G-Industrial-Computer/>. [cit. 2023-12-06].
- [18] ETSI TS 123 501 V15.2.0 (2018-06) *5G; System Architecture for the 5G System (3GPP TS 23.501 version 15.2.0 Release 15)* Online: v Brně, 2023. Dostupné z: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/15.02.00_60/ts_123501v150200p.pdf. [cit. 2023-11-20].
- [19] TECHTARGET *5G NSA vs. SA: How do the deployment modes differ?* Online: v Brně, 2023. Dostupné z: <https://www.techtarget.com/searchnetworking/feature/5G-NSA-vs-SA-How-does-each-deployment-mode-differ>. [cit. 2023-12-05].
- [20] ERICSSON *5G migration strategy from EPS to 5G system* Online: v Brně, 2023. Dostupné z: <https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/migration-from-eps-to-5gs>. [cit. 2023-11-21].

- [21] KEYSIGHT *Understanding 5G New Radio Bandwidth Parts* Online: v Brně, 2023. 2018-10-31 Dostupné z: <https://www.keysight.com/blogs/inds/2018/10/31/understanding-5g-new-radio-bandwidth-parts>. [cit. 2023-12-06].
- [22] MONIEM-TECH *What is Control and User Plane Separation (CUPS)* Online: v Brně, 2023. Dostupné z: <https://moniem-tech.com/2022/11/30/what-is-control-and-user-plane-separation-cups/>. [cit. 2023-11-14].
- [23] XIAOLI ZHANG, YUE LAI a SHENGLI XIE. *A GTP stateful inspection method based on network processor* Online: v Brně, 2024. Dostupné z: <https://ieeexplore.ieee.org/document/4737332>. [cit. 2024-05-20].
- [24] DOCKER *Docker overview* Online: v Brně, 2024. Dostupné z: <https://docs.docker.com/get-started/overview/>. [cit. 2024-02-27].
- [25] VITE *Getting Started* Online: v Brně, 2024. Dostupné z: <https://vitejs.dev/guide/>. [cit. 2024-02-27].
- [26] YARN *Introduction* Online: v Brně, 2024. Dostupné z: <https://yarnpkg.com/getting-started>. [cit. 2024-02-28].
- [27] ESLINT *Getting Started with ESLint* Online: v Brně, 2024. Dostupné z: <https://eslint.org/docs/latest/use/getting-started>. [cit. 2024-03-03].
- [28] PRETTIER *Why Prettier?* Online: v Brně, 2024. Dostupné z: <https://prettier.io/docs/en/why-prettier>. [cit. 2024-03-03].
- [29] GIT *Getting Started - What is Git?* Online: v Brně, 2024. Dostupné z: <https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F>. [cit. 2024-03-03].
- [30] GITHUB *About GitHub and Git* Online: v Brně, 2024. Dostupné z: <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>. [cit. 2024-03-03].
- [31] META *Quick Start React* Online: v Brně, 2024. Dostupné z: <https://react.dev/learn>. [cit. 2024-03-17].
- [32] MICROSOFT *TypeScript for the New Programmer TypeScript* Online: v Brně, 2024. Dostupné z: <https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>. [cit. 2024-03-17].
- [33] SASS *Documentation* Online: v Brně, 2024. Dostupné z: <https://sass-lang.com/documentation/>. [cit. 2024-03-17].

- [34] GETBEM *Introduction* Online: v Brně, 2024. Dostupné z: <https://getbem.com/introduction/>. [cit. 2024-04-27].
- [35] TAILWIND *Rapidly build modern websites without ever leaving your HTML*. Online: v Brně, 2024. Dostupné z: <https://tailwindcss.com/>. [cit. 2024-03-17].
- [36] ZOD *Introduction* Online: v Brně, 2024. Dostupné z: <https://zod.dev/>. [cit. 2024-04-10].
- [37] FORMIUM *Overview Formik docs* Online: v Brně, 2024. Dostupné z: <https://formik.org/docs/overview>. [cit. 2024-04-10].
- [38] TANSTACK QUERY *Overview* Online: v Brně, 2024. Dostupné z: <https://tanstack.com/query/latest/docs/framework/react/overview>. [cit. 2024-04-10].
- [39] WEBKID GMBH *Introduction React Flow* Online: v Brně, 2024. Dostupné z: <https://reactflow.dev/learn/concepts/introduction>. [cit. 2024-04-15].
- [40] RASCASONE *UX A UI DESIGN: JAK NA UŽIVATELSKÉ ROZHRANÍ WEBŮ A APLIKACÍ?* Online: v Brně, 2023. Dostupné z: <https://www.rascasone.com/cs/blog/ux-design-ui-design>. [cit. 2023-12-08].

Zoznam symbolov a skratiek

3GPP	The 3rd Generation Partnership Project
GSM	Global System for Mobile Communications
UMTS	The Universal Mobile Telecommunications System
SDOs	Standards Development Organizations
UTRA	Universal Terrestrial Radio Access
FDD	Frequency Division Duplex
TDD	Time Division Duplex
IoT	Internet of Things
M2M	Machine-to-Machine
mmWave	millimeter Wave
MIMO	multiple input - multiple output
Core	Core Network
Backbone	Backbone Network
RAN	Radio Access Network
TTI	Time Transmit Intervals
NF	Network Function
NFV	Network Functions Virtualization
SDN	Software Defined Networking
DRA	Diameter Routing Agent
UP	User Plane
CP	Control Plane
AMF	Access and Mobility Management Function
SMF	Session Management Function
PGW	Packet Data Network Gateway

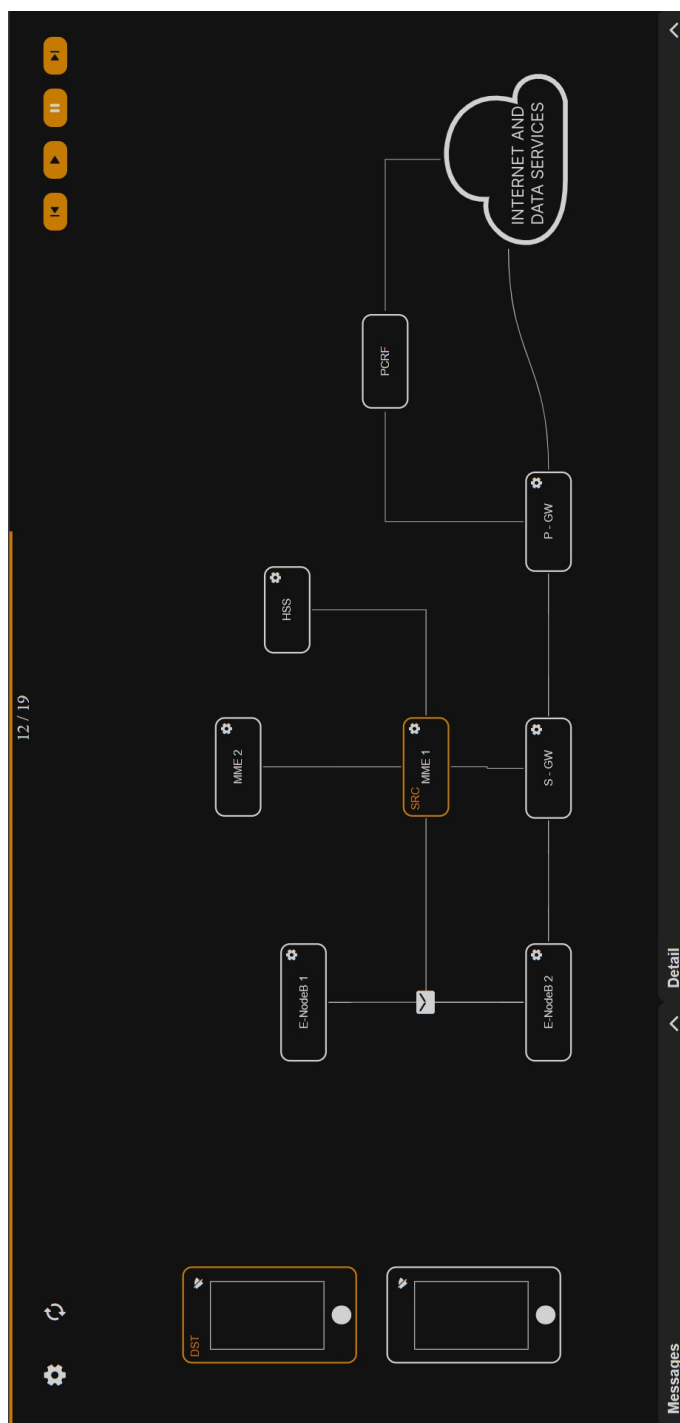
PGW-C	Packet Data Network Gateway Control Plane
PGW-U	Packet Data Network Gateway User Plane
MME	Mobility Management Entity
PDU	Protocol Data Unit
PDN	Packet Data Network
EPS	Evolved Packet System
MNO	Mobile Network Operator
FCC	Federal Communications Commission
CSPs	Communications Service Providers
GSMA	Groupe Spéciale Mobile Association
LTE	Long Term Evolution
5G NR	5G New Radio
SA	Standalone
NSA	Non Standalone
EPC	Evolved Packet Core
mMTC	Massive Machine-Type Communications
eMBB	Enhanced Mobile Broadband
URLLC	Ultra-Reliable Low-Latency Communication
UPF	User Plane Function
OFDM	Orthogonal Frequency Division Multiplexing
MT	Mobile Termination
TE	Terminal Equipment
ME	Mobile Equipment
UICC	Universal Integrated Circuit Card
USIM	Universal Subscriber Identity Module

E-UTRAN	Evolved Universal Terrestrial Radio Access Network
HSS	Home Subscriber Server
S-GW	Serving Gateway
P-GW	Packet Data Network Gateway
PCRF	Policy and Charging Rules Function
UX	User Experience
UI	User Interface
GPRS	General Packet Radio Service
EDGE	Enhanced Data Rates for GSM Evolution
ISM	Industrial, Scientific, and Medical
FE	Frontend
BE	Backend
RACH	Random Access Channel
AR	Augmented Reality
ETWS	Earthquake and Tsunami Warning System
EIR	Equipment Identity Register
QoS	Quality of Service
NFV	Network Functions Virtualization
N3IWF	non-3GPP Inter-Working Functions
GTP	GRPS Tunnelling Protocol
API	Application Programming Interface

Zoznam príloh

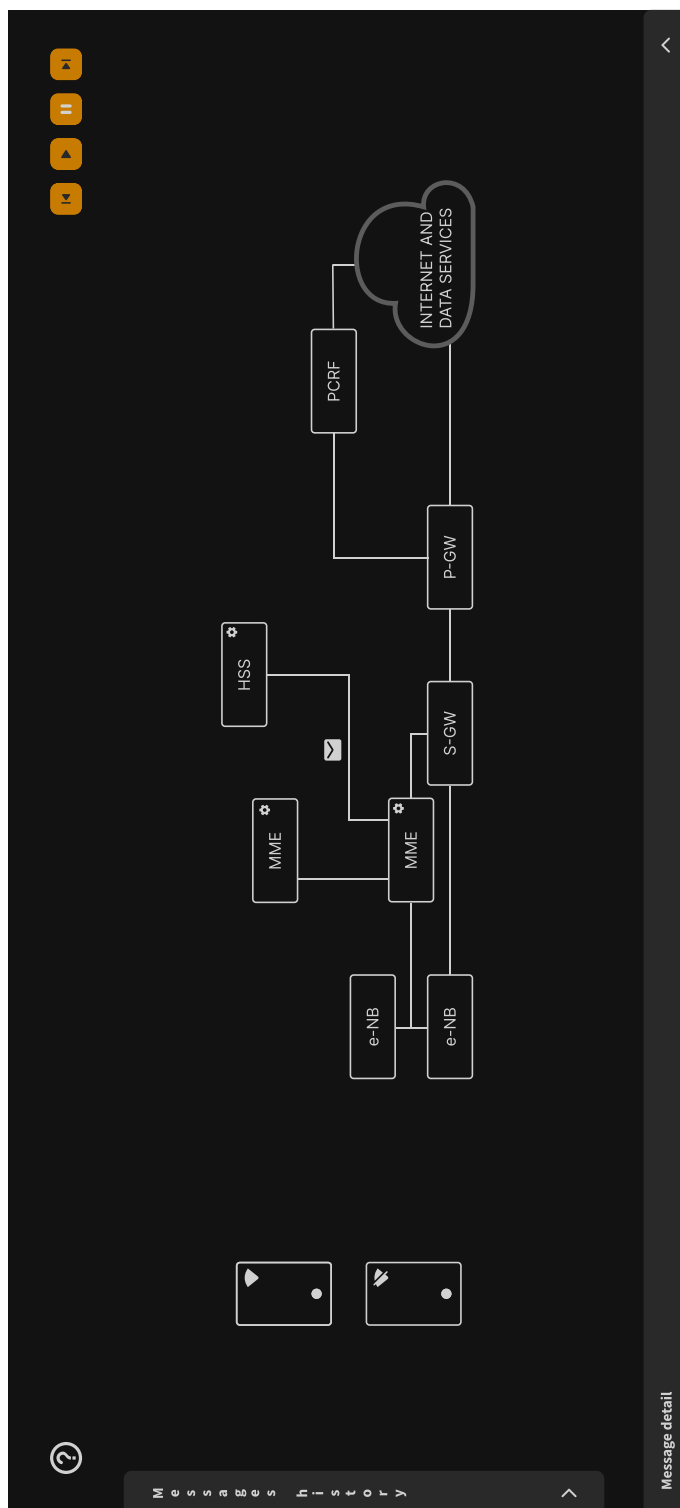
A Aplikácia	70
B Návrh UI aplikácie	71
C UI história správ	72
D UI detail správy	73
E UI modal okno	74
F Komponenty UI 1	75
G Komponenty UI 2	76
H Elektronická príloha	77

A Aplikácia



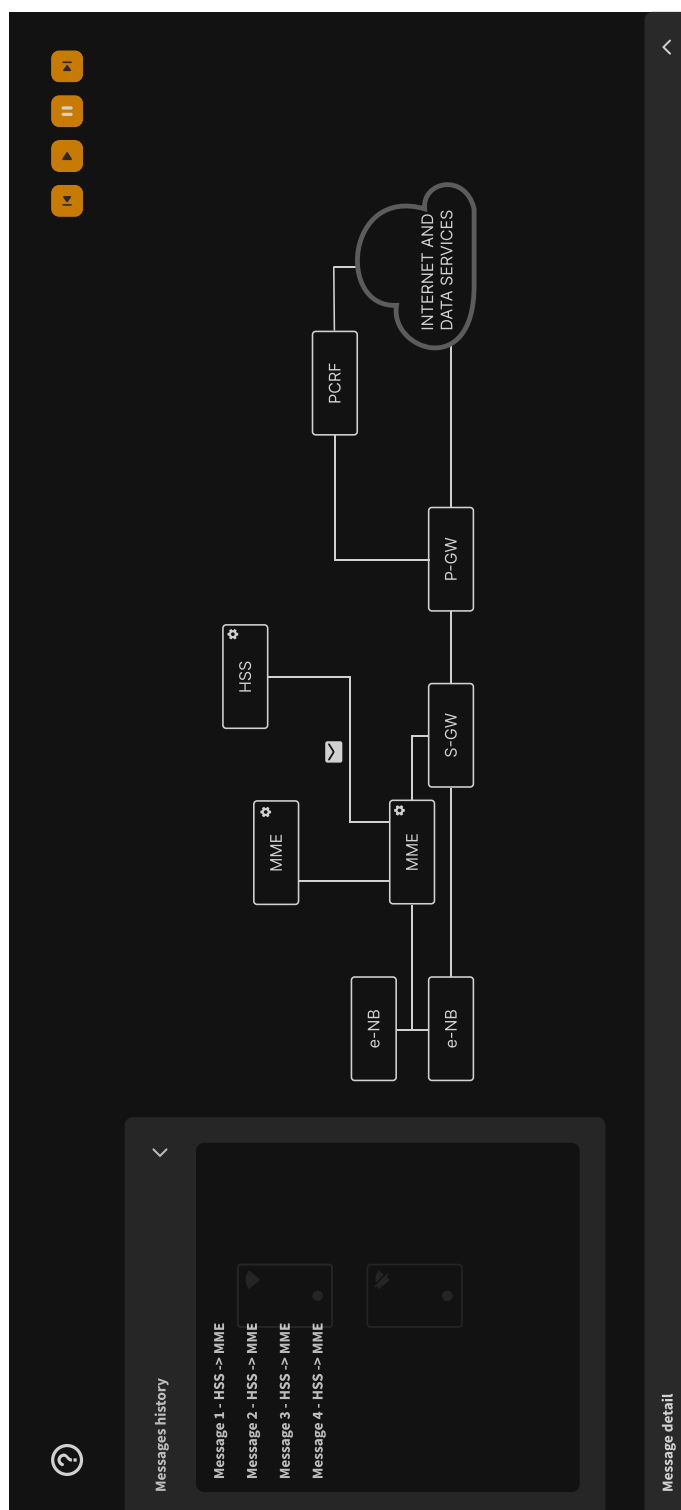
Obr. A.1: Aplikácia

B Návrh UI aplikácie



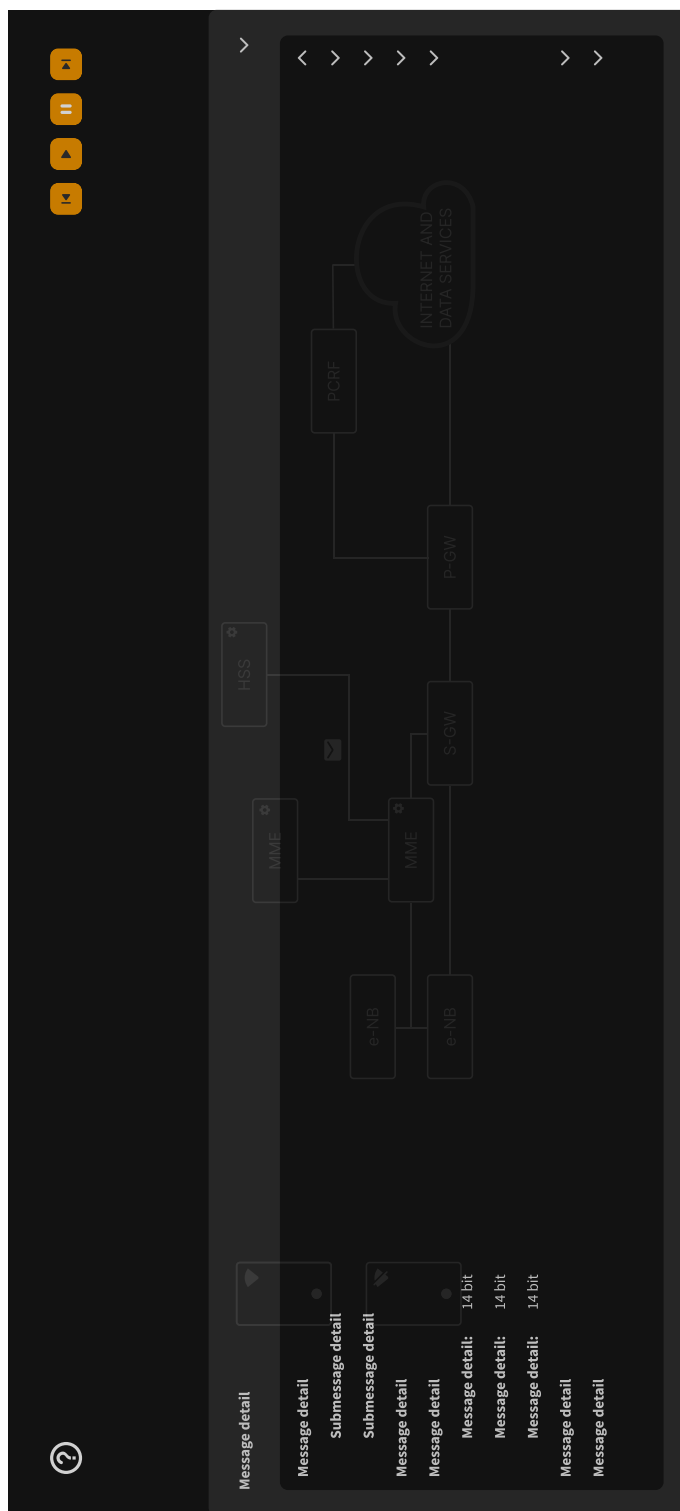
Obr. B.1: Grafický návrh aplikácie

C UI história správ



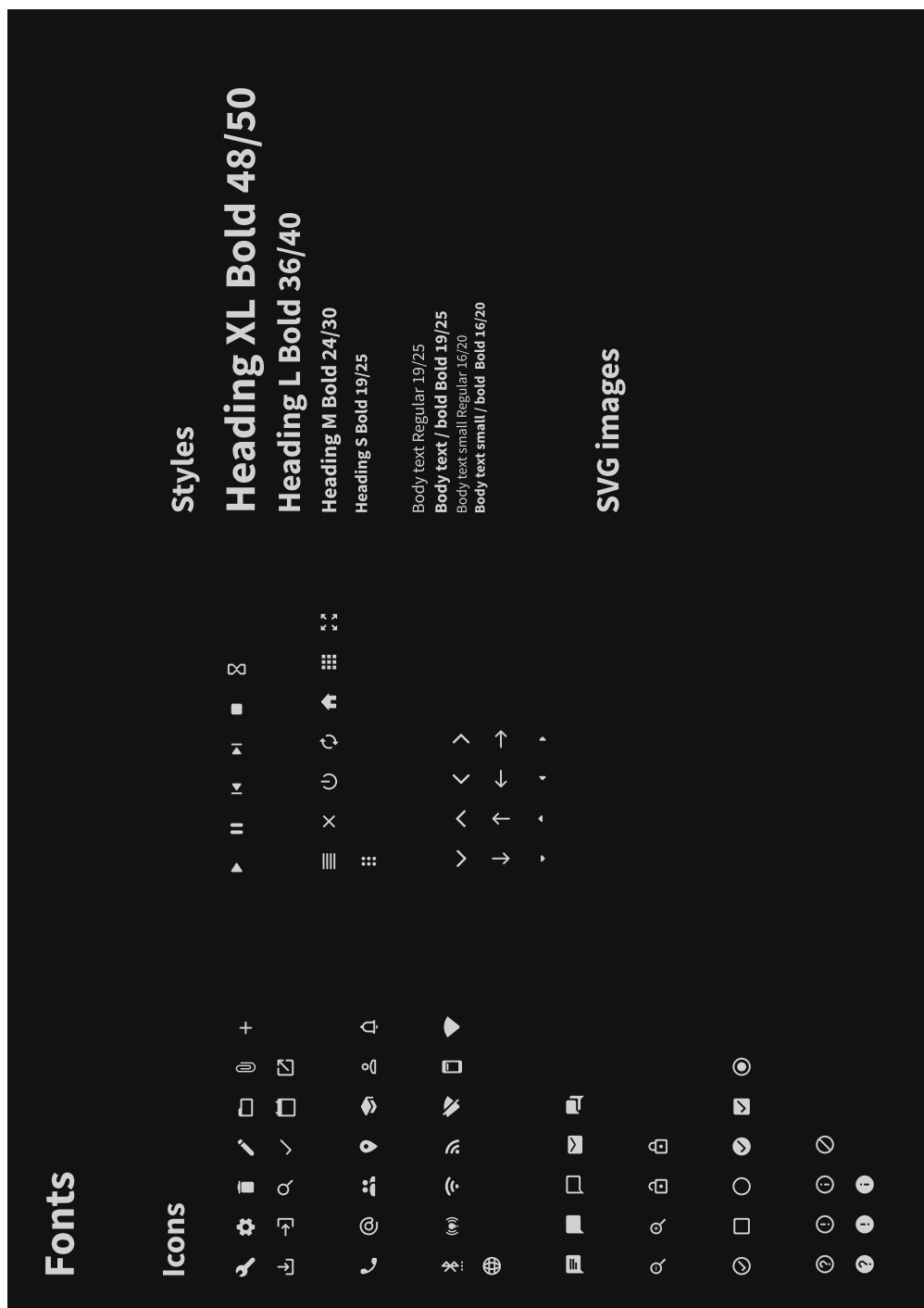
Obr. C.1: Grafický návrh aplikácie - história správ

D UI detail správy



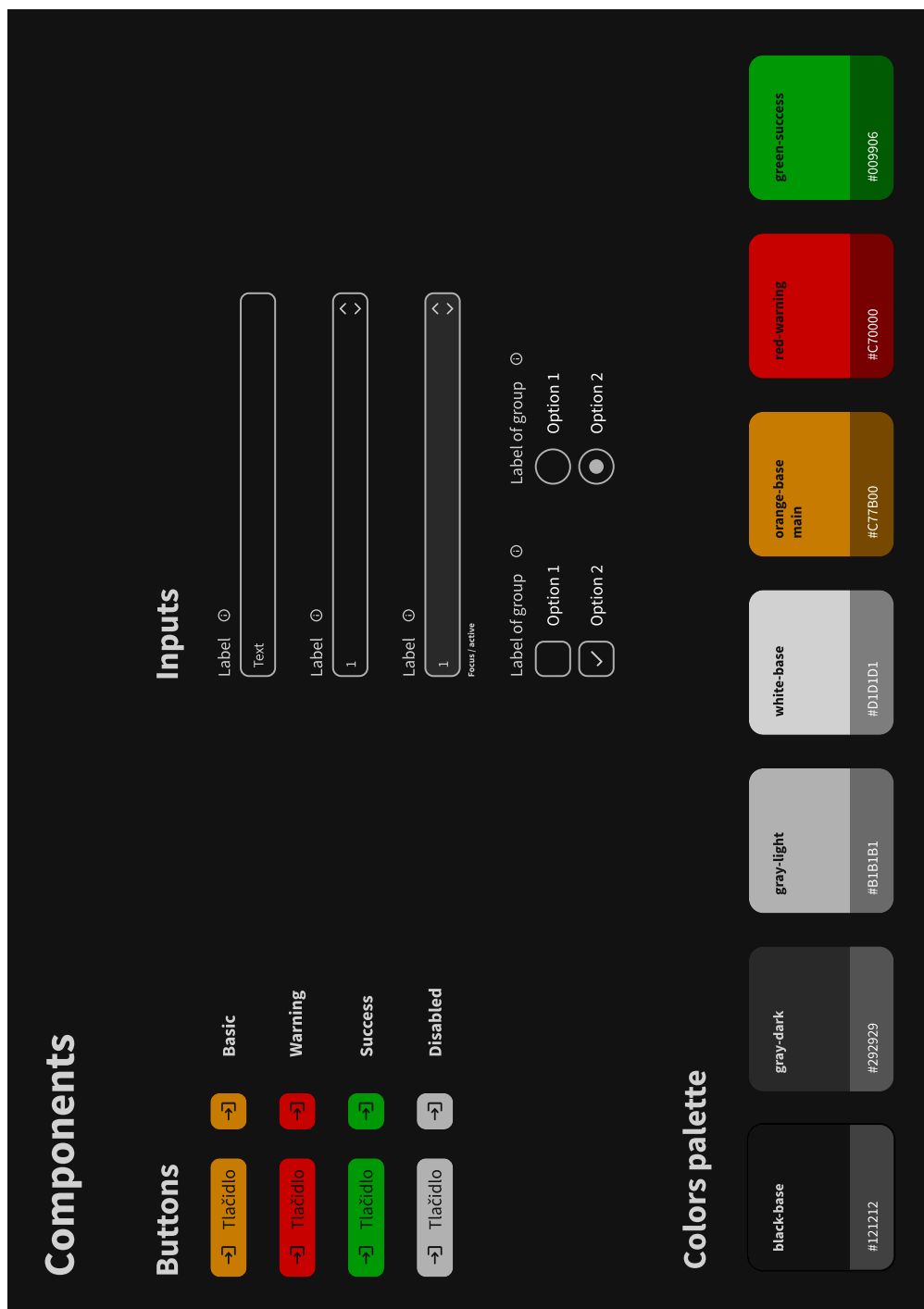
Obr. D.1: Grafický návrh aplikácie - detail správy

F Komponenty UI 1



Obr. F.1: Komponenty aplikácie 1

G Komponenty UI 2



Obr. G.1: Komponenty aplikácie 2

H Elektronická príloha

```
/ ..... koreňový adresár priloženého archívu
├── app ..... adresár aplikácie
│   ├── .vscode
│   │   ├── extensions.json
│   │   └── settings.json
│   ├── @types
│   │   ├── formik.d.ts
│   │   ├── style.d.ts
│   │   └── vite.d.ts
│   ├── src
│   │   ├── font ..... obsahuje fonty pre ikony
│   │   ├── images ..... obsahuje obrázky pre aplikáciu
│   │   ├── js
│   │   │   ├── components ..... obsahuje menšie komponenty
│   │   │   ├── features ..... obsahuje rôzne nástroje a funkcie
│   │   │   ├── layout ..... obsahuje komponenty s hlavným rozložením
│   │   │   ├── models ..... obsahuje zod schémy a typy
│   │   │   └── nodes ..... obsahuje prvky siete s nastaveniami
│   │   ├── scss
│   │   │   ├── base ..... obsahuje základné štýly a reset
│   │   │   ├── components ..... obsahuje štýly pre komponenty
│   │   │   ├── layout ..... obsahuje štýly pre rozloženie
│   │   │   ├── utils ..... obsahuje rôzne funkcie pre Sass
│   │   │   └── app.scss
│   │   └── client.tsx
│   ├── .dockerignore
│   ├── .env
│   ├── .eslintrc.json
│   ├── .gitignore
│   ├── .prettierrc.json
│   ├── .app.Dockerfile
│   ├── .docker-compose.yml
│   ├── .index.html
│   ├── .package.json
│   ├── .postcss.config.js
│   ├── .README.md
│   ├── .tailwind.config.js
│   ├── .tsconfig.json
│   ├── .vite.config.ts
│   └── .yarn.lock
├── figma_design.fig ..... súbor programu figma (UI)
├── diplomova-praca.code-workspace ..... workspace súbor pre VS Code
└── README.md
```