Vpeljava zabojnikov v oblačno zasnovana zasebna omrežja 5G

Urban Zaletel, Kristjan Voje, Benjamin Burgar, Uroš Brovč

Kontron d.o.o, Kranj, Slovenija
urban.zaletel@kontron.si, kristjan.voje@kontron.si,
benjamin.burgar@kontron.si, uros.brovc@kontron.si

Današnja sodobna podjetja imajo vse večje zahteve na področju mobilne povezljivosti. Nastajajoče aplikacije in industrijska okolja zahtevajo večjo pasovno širino, predvsem pa izjemno nizko zakasnitev, več spektra in večjo zanesljivost. Z dodeljevanjem namenskega spektra za ta okolja s strani regulatorjev (Nemčija, Norveška, Združeno kraljestvo in druge napredne države) so v podjetjih nastali pogoji za uvajanje zasebnih mobilnih omrežij, ki so samostojna in razširjajo njihova obstoječa brezžična omrežja. Oblačno zasnovana omrežja podjetjem omogočajo učinkovito upravljanje in pomagajo pri sprejemanju hitrejših in boljših poslovnih odločitev. V prispevku smo naslovili motivacijo prehoda na oblačno arhitekturo in pridobljene izkušnje, ki smo jih pridobili ob razvoju in uvajanju oblačno zasnovanih rešitev na Kubernetesu. Razvoj 5G je zahteval usklajenost med ekipami, ki so razvijali omrežne elemente v različnih jezikih (Java, Golang, C) in enoten pristop do modernih DevSecOps praks (infrastruktura-kot-koda, avtomatizacija, varnostno skeniranje, Helm). Kubernetes in mikroservisi so nov koncept za telekomunikacijske aplikacije, ki se počasi selijo iz paradigme VNF (virtual network functions) proti paradigmi CNF (cloud-native network functions)[6].
V prispevku so izpostavljeni izzivi, s katerimi smo se soočali pri implementacij oblačnih omrežnih funkcij. Nanizane so tudi številne pridobljene kompetence, ki smo jih inženirji pridobili tekom trajanja projekta.

Ključne besede:
5G
privatna omrežja
CNF
DevOps
DevSecOps
Kubernetes
Helm

Univerzitetna založba Univerze v Mariboru
ISBN 978-961-286-772-0
DOI https://doi.org/10.18690/um.feri.8.2023.5

1 Uvod

5G privatna omrežja predstavljajo razširitev 5G tehnologije na področje podjetij in organizacij, ki želijo vzpostaviti lastno zasebno omrežje z visoko zmogljivostjo in nizko zakasnitvijo. Ta omrežja omogočajo podjetjem večji nadzor, varnost in prilagodljivost pri povezovanju njihovih naprav in sistemov.

Privatna 5G omrežja se gradijo na enakih temeljih kot javna 5G omrežja, vendar se upravljajo in nadzirajo znotraj organizacije. To pomeni, da podjetja dobijo lastno omrežno infrastrukturo, vključno s baznimi postajami in strežniškimi zmogljivostmi, kar jim omogoča, da prilagodijo omrežje svojim specifičnim potrebam.

Eden glavnih razlogov za uvedbo privatnih 5G omrežij je izboljšanje zanesljivosti in varnosti povezave. Podjetja lahko nadzorujejo in omejijo dostop do svojega omrežja, kar zmanjšuje tveganje za morebitne vdore ali izpade povezave. Poleg tega imajo privatna omrežja višjo prioriteto in večjo pasovno širino, kar zagotavlja stabilno povezavo za kritične aplikacije.

Privatna 5G omrežja omogočajo tudi večjo fleksibilnost pri upravljanju in integraciji naprav. Organizacije lahko povežejo različne naprave in senzorje v svoje omrežje ter izkoristijo prednosti interneta stvari. To odpira vrata za inovacije na področjih kot so industrija 4.0, pametne tovarne, avtonomna vozila, zdravstvena oskrba in še veliko več.

Kljub temu pa obstajajo tudi nekateri izzivi pri uvedbi privatnih 5G omrežij, vključno z visokimi stroški vzpostavitve in vzdrževanja infrastrukture ter potrebo po posebnem znanju in usposabljanju za upravljanje omrežja.

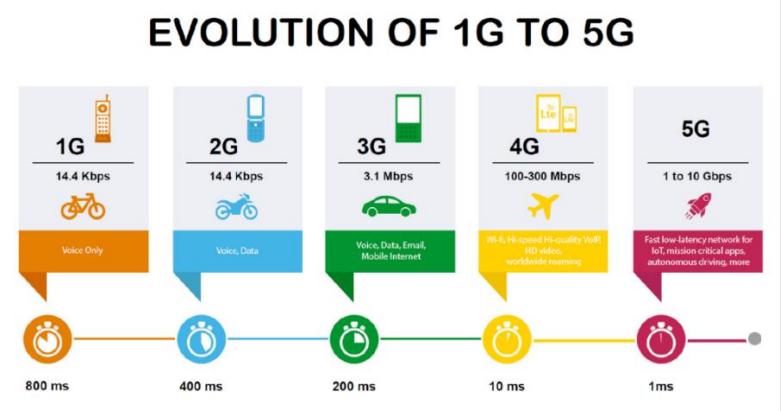
Optimizacija stroškov postavitve in vzdrževanja 5G privatnih omrežij se lahko doseže s pomočjo uporabe oblačnih postavitev. Oblačno zasnovana omrežja omogočajo organizacijam koriščenje infrastrukture zunanjih ponudnikov oblakov, kar je v nekaterih primerih veliko bolj učinkovito od postavljanja in vzdrževanja svoje infrastrukture.

2 5G omrežje

2.1 Evolucija mobilnih omrežij

Evolucija mobilnih omrežij po generacijah je temeljila na potrebi po zadovoljevanju naraščajoče potrebe po prenosu podatkov, izboljšanju učinkovitosti omrežja in omogočanju novih storitev in aplikacij. Vsaka generacija je nadgradila prejšnjo, uvedla nove tehnologije in zmogljivosti, ki so oblikovale način komunikacije in interakcij v digitalni dobi.

Na sliki 1 so podane ključne značilnosti posamezne generacije mobilnih omrežij.



Slika 1: Evolucija mobilnih omrežij
Vir: [3].

2.2 Kaj nam omogoča 5G

5G omrežje omogoča izjemno hitrost prenosa podatkov, nizko zakasnitev ter večjo zmogljivost in povezljivost, kar omogoča napredne aplikacije, inovacije ter nove poslovne modele v različnih panogah.

Pregled ključnih prednosti:

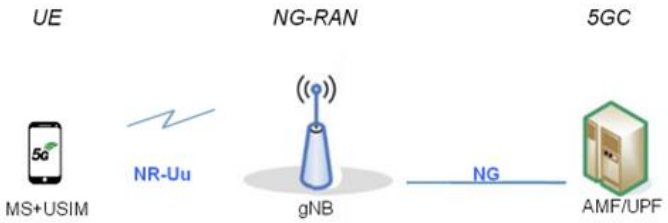
- Visoka hitrost prenosa podatkov: izjemno visoke hitrosti prenosa podatkov, ki presegajo zmogljivosti prejšnjih generacij omrežij. To vključuje podporo za prenos velikih količin podatkov, kot so visokokakovostni video posnetki in pretočne storitve v realnem času.
- Nizka zakasnitev (latenca): izjemno nizka zakasnitev pri prenosu podatkov, kar je ključno za aplikacije, ki zahtevajo takojšen odziv. To vključuje avtonomna vozila, virtualno resničnost, obogateno resničnost in druge aplikacije, ki zahtevajo hitro in brezhibno delovanje v realnem času.
- Večja zmogljivost omrežja: zasnova za večjo zmogljivost, kar pomeni, da mora omogočati hkratno povezavo večjega števila naprav na istem območju. To je ključno za podporo različnim napravam v okviru interneta stvari (IoT), kot tudi za zagotavljanje visoke kakovosti storitev v gostih urbanih območjih.
- Podpora za različne uporabniške scenarije: prilagodljivost in sposobnost podpirati različne uporabniške scenarije, vključno z mobilnimi napravami, IoT napravami, industrijsko avtomatizacijo, pametnimi mesti in drugimi naprednimi aplikacijami.
- Energijska učinkovitost: zmanjšuje porabo energije v omrežju in podaljšuje življenjsko dobo baterij pri napravah.
- Visoka zanesljivost in varnost: zagotavlja visoko zanesljivost in varnost komunikacije, še posebej pri kritičnih aplikacijah, kot so kritična infrastruktura, pametni sistemi v prometu in oddaljena medicinska oskrba. Omrežje mora imeti tudi ustrezne varnostne mehanizme za zaščito uporabniških podatkov in preprečevanje napadov.

2.3 Arhitektura

Arhitektura 5G omrežja je bila sestavljena tako, da zagotavlja zahteve, kot so zapisane v poglavju 2.2. Ključne komponente 5G arhitekture vključujejo (slika 2):

- Uporabniška oprema[1] (UE): Naprava, ki jo uporabnik uporablja za dostop do 5G omrežja, na primer pametni telefon, tablica ali druga naprava z vgrajeno 5G tehnologijo.
- Radijsko dostopno omrežje[2] (RAN): Brezžično omrežje, ki omogoča komunikacijo med UE in omrežno infrastrukturo. RAN v 5G sistemu je znan kot NG-RAN in uporablja tehnologije, kot so Massive MIMO[3] in oblikovanje snopa[4] [7] za izboljšanje zmogljivosti in kakovosti povezave.
- 5G jedro[5] (5GC): Osrednji del arhitekture omrežja 5G, ki obvladuje in upravlja vse ključne funkcije omrežja. Deluje kot centralni vozliščni sistem, ki zagotavlja povezljivost, komunikacijo in upravljanje med uporabniki, napravami in storitvami v omrežju 5G.

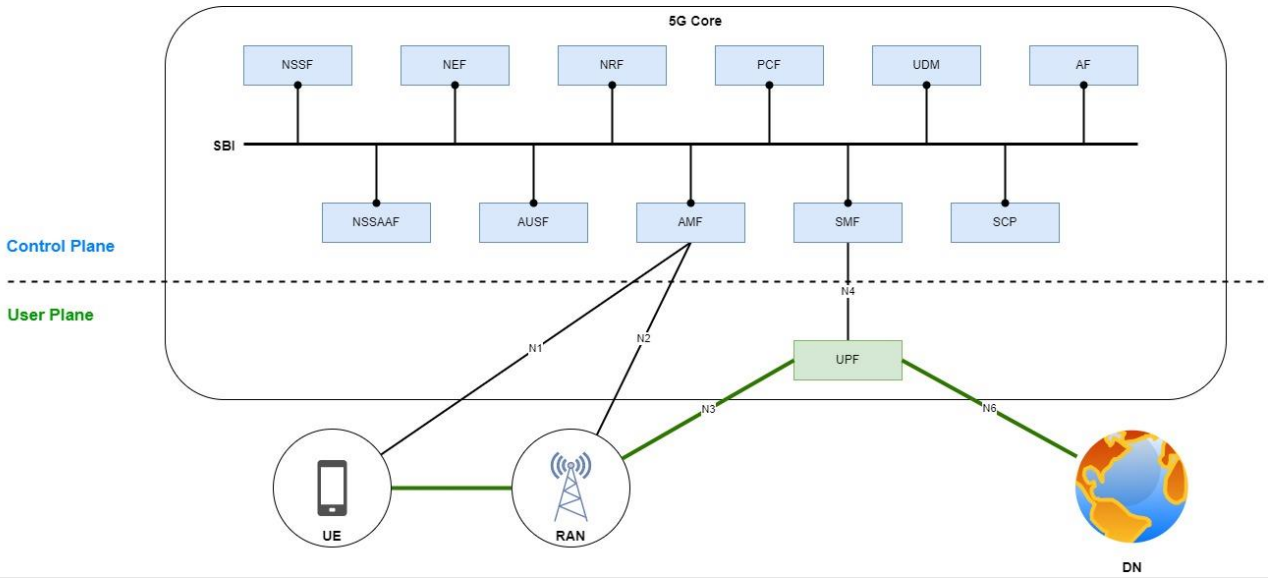
[1] uporabniška oprema – izvorno "user equipment", okrajšava "UE"
[2] radijsko dostopno omrežje - izvorno "radio access network", okrajšava "RAN"
[3] mMIMO – Massive Multiple Input Multiple Output
[4] oblikovanje snopa – izvorno "beamforming"
[5] 5G jedro – izvorno "5G core", okrajšava "5GC"



Slika 2: Pregled 5G sistema
Vir: [1].

2.4 Omrežne funkcije in vmesniki

Podrobnejša arhitektura, z osnovnimi omrežnimi funkcijami[6] jedra (NF), vmesniki med omrežnimi funkcijami, radijskim delom (RAN in UE) ter podatkovnim omrežjem[7] (DN), je prikazana na sliki 3.



Slika 3: Arhitektura 5G sistema.

Arhitektura 5G omrežja je specificirana v standardu 3GPP[8]. Jedro 5G omrežja oz. omrežne funkcije lahko v osnovi delimo na kontrolno ravnino[9] in uporabniško ravnino[10].

Kontrolna ravnina je del omrežja, ki upravlja in nadzira pretok podatkov med napravami in storitvami. Nekatere od njenih nalog so:

- **Alokacija virov**: kontrolna ravnina dodeljuje frekvenčne pasove, kanale in časovne reže za prenos podatkov med napravami in baznimi postajami.
- **Upravljanje mobilnosti**: kontrolna ravnina skrbi za nemoteno preklapljanje med različnimi celicami in omrežji, ko se naprave premikajo po prostoru.
- **Upravljanje povezljivosti**: kontrolna ravnina vzpostavlja, vzdržuje in prekinja povezave med napravami in storitvami, glede na njihove potrebe in zahteve.

[6] omrežna funkcija – izvorno "Network Function", okrajšava "NF"
[7] podatkovno omrežje - izvorno "data network", okrajšava "DN"
[8] 3GPP - 3rd Generation Partnership Project [2]
[9] kontrolna ravnina – izvorno "control plane"
[10] uporabniška ravnina – izvorno "user plane"

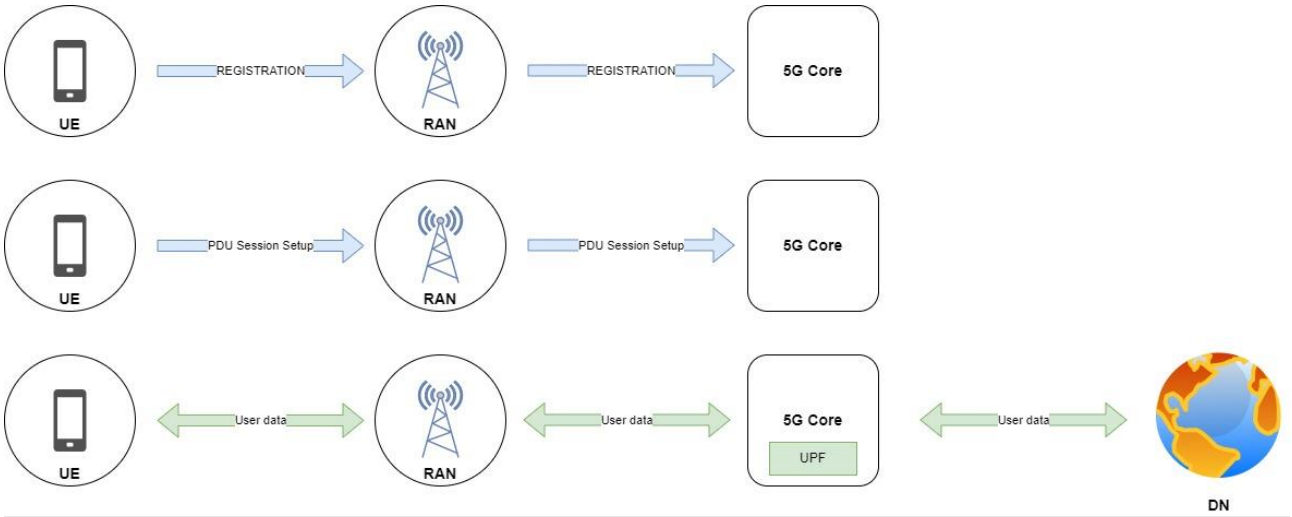
- **Upravljanje identitete**: kontrolna ravnina preverja avtentičnost in avtorizacijo naprav in uporabnikov, ki dostopajo do omrežja in storitev.
- **Upravljanje varnosti**: kontrolna ravnina zagotavlja šifriranje, integriteto in zaupnost podatkov, ki potekajo po omrežju.

Uporabniška ravnina je del omrežja, ki prenaša in usmerja podatke med napravami in storitvami. Nekatere od njenih nalog so:

- **Prenos podatkov:** Omogoča učinkovit prenos podatkov med uporabniškimi napravami in omrežjem. To vključuje prenos glasovnih klicev, videokonferenc, prenosa datotek, internetnega brskanja in drugih oblik podatkovne komunikacije.
- **Usmerjanje in preklapljanje**: Zagotavlja usmerjanje podatkovnega prometa in preklapljanje med omrežnimi vozlišči za optimalno pot prenosa podatkov.
- **Nadzor kakovosti storitev**[11]: Skrbi za nadzor in upravljanje kakovosti storitev. To pomeni, da se podatkovni promet razvrsti in obdela glede na različne parametre, kot so pasovna širina, zakasnitev, izgube paketov in druge metrike, da se zagotovi ustrezna raven storitev in izboljša uporabniška izkušnja.
- **Šifriranje in varnost:** Odgovorna je za šifriranje podatkov in zagotavljanje varnosti med prenosom, da se zaščiti pred nepooblaščenim dostopom.
- **Komunikacija z omrežnimi storitvami**: Sodeluje z omrežnimi funkcijami za avtentikacijo, identifikacijo, upravljanje mobilnosti in nadzor dostopa.

2.5 Registracija in vzpostavitev seje

5G kompatibilna naprava z ustrezno SIM kartico, ki želi dostopati do podatkovnega omrežja (DN) preko 5G omrežja, se mora najprej prijaviti v 5G omrežje. Uspešna prijava naprave je predpogoj za vzpostavitev PDU seje[12] do 5G jedra. PDU seja je logična povezava med uporabniško napravo (UE) in 5G jedrom (natančneje med UE in UPF[13]), ki uporabniku omogoča povezavo z podatkovnim omrežjem (DN). Preko PDU seje se omogoči pretok uporabniškega prometa med 5G napravo (UE) in ciljno storitvijo (npr. brskanje po internetu ali pošiljanje senzorskih meritev na IoT platformo). Osnovni koraki vzpostavitve PDU seje so prikazani na sliki 4.



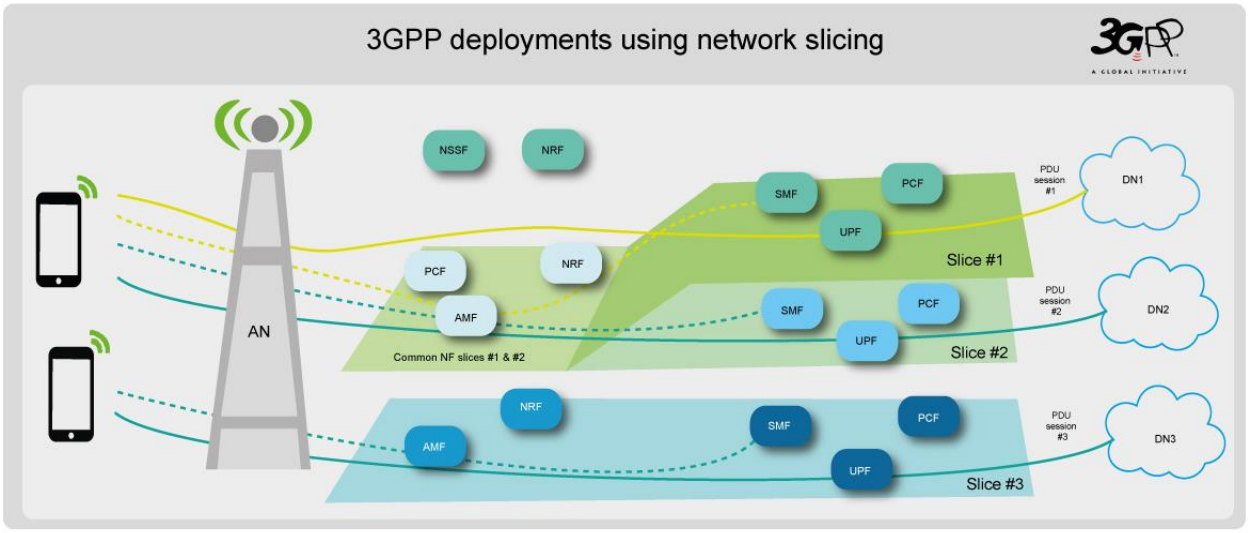
Slika 4: Vzpostavitev PDU seje.

[11] kakovost storitve – izvorno "Quality of Service", okrajšava "QoS"
[12] PDU seja – izvorno "Protocol Data Unit Session", okrajšava "PDU session"
[13] UPF – izvorno "User Plane Function"

2.6 Segmentacija 5G omrežja in kakovost storitev

Med glavne in kompleksnejše funkcionalnosti 5G omrežja štejemo segmentacijo in upravljanje s kakovostjo storitev.

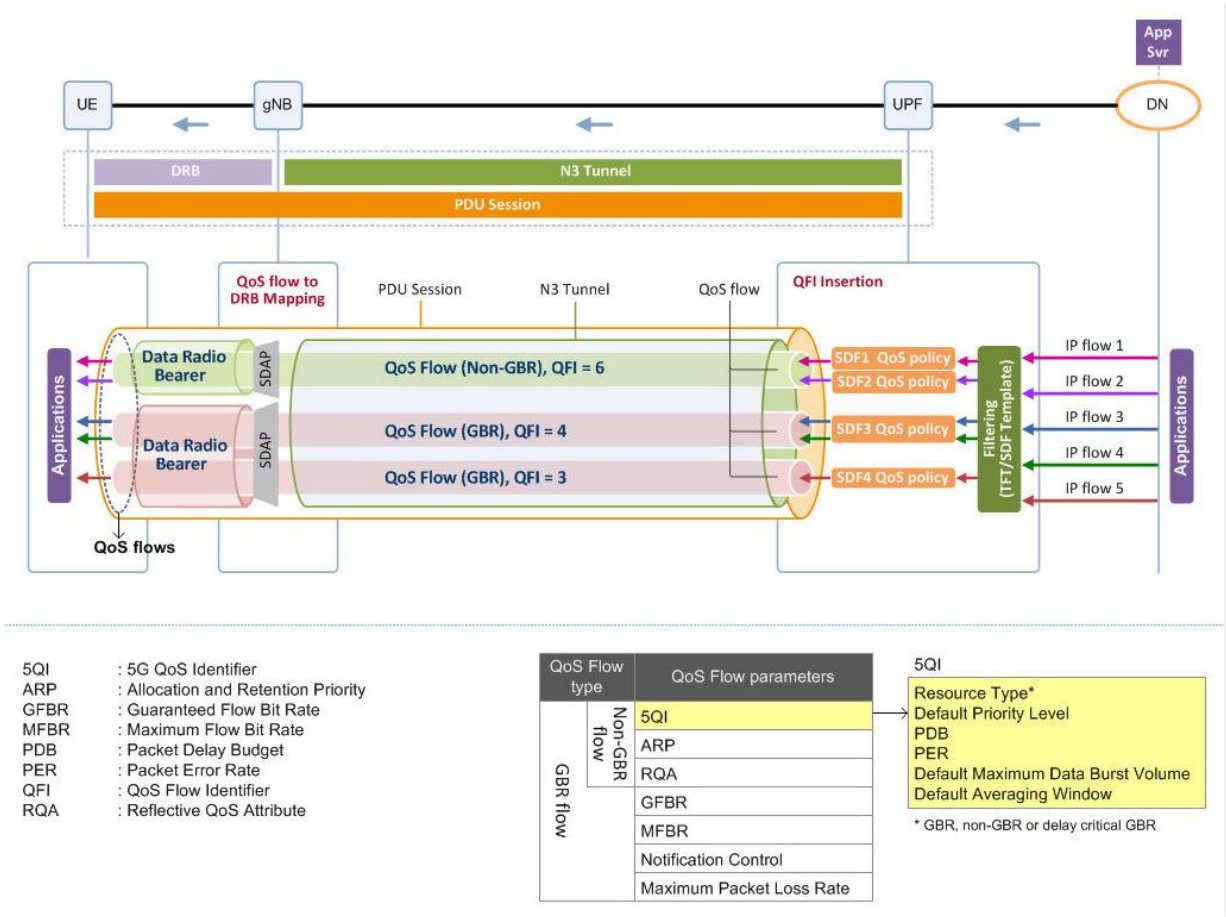
Segmentacija 5G omrežja[14] je koncept, ki omogoča segmentacijo omrežja 5G na več virtualnih omrežij, imenovanih "rezine"[15]. Vsaka rezina je samostojno virtualno omrežje z lastnimi zmogljivostmi in parametri kakovosti storitve[16] (QoS). Rezine omogočajo prilagojeno in optimizirano povezljivost za različne vrste uporabnikov, aplikacij in industrijskih vertikal. S segmentacijo 5G omrežja se omogoča izpolnjevanje specifičnih zahtev glede zmogljivosti, odzivnega časa, zanesljivosti in varnosti, kar omogoča učinkovito upravljanje različnih uporabniških scenarijev v okviru enega samega omrežja. Rezine se lahko dinamično konfigurira in prilagaja, kar omogoča prilagajanje omrežnih virov glede na spreminjajoče se zahteve in potrebe uporabnikov. Na sliki 5 je primer omrežja s segmentacijo.



Slika 5: Segmentacija 5G omrežja
Vir: [4].

Kakovost storitev (QoS) je koncept, ki se nanaša na sposobnost omrežja 5G, da zagotavlja različne ravni storitev in zadovolji določene zahteve glede kakovosti in učinkovitosti prenosa podatkov (Slika 6). S pomočjo QoS lahko omrežje prilagodi svoje delovanje glede na specifične potrebe aplikacij, uporabnikov in storitev. To omogoča, da se različni vrsti prometa dodelijo prednostne ravni, uporabljajo se mehanizmi za prioritetno obdelavo prometa ter zagotavlja se kakovostno in zanesljivo izkušnjo uporabnikom.

[14] segmentacija 5G omrežja - izvorno "5G slicing"
[15] rezina – izvorno "slice"
[16] kakovost storitve – izvorno "Quality of Service", okrajšava "QoS"



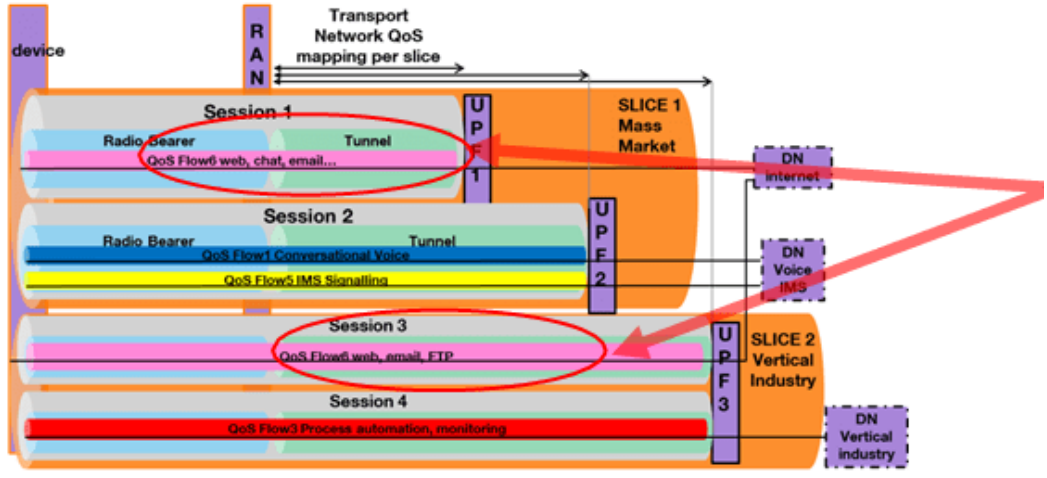
Slika 6: QoS 5G omrežja

Vir: [5].

QoS lahko razlikuje med različnimi vrstami prometa, vendar ne more razlikovati in različno obravnavati iste vrste prometa, ki prihaja iz različnih virov. QoS nima sposobnosti za izvedbo izolacije prometa od konca do konca.

Po drugi strani 5G segmentacija skupaj s kvaliteto storitev (QoS), lahko razlikuje med istimi vrstami prometa, ki prihajajo od različnih najemnikov/rezin. Segmentacija omrežja omogoča delovanje več logičnih omrežij na osnovi skupne fizične omrežne infrastrukture, kar zagotavlja izolacijo med njimi.

Na sliki 7 je primer združevanja koncepta segmentacije in kvalitete storitev.

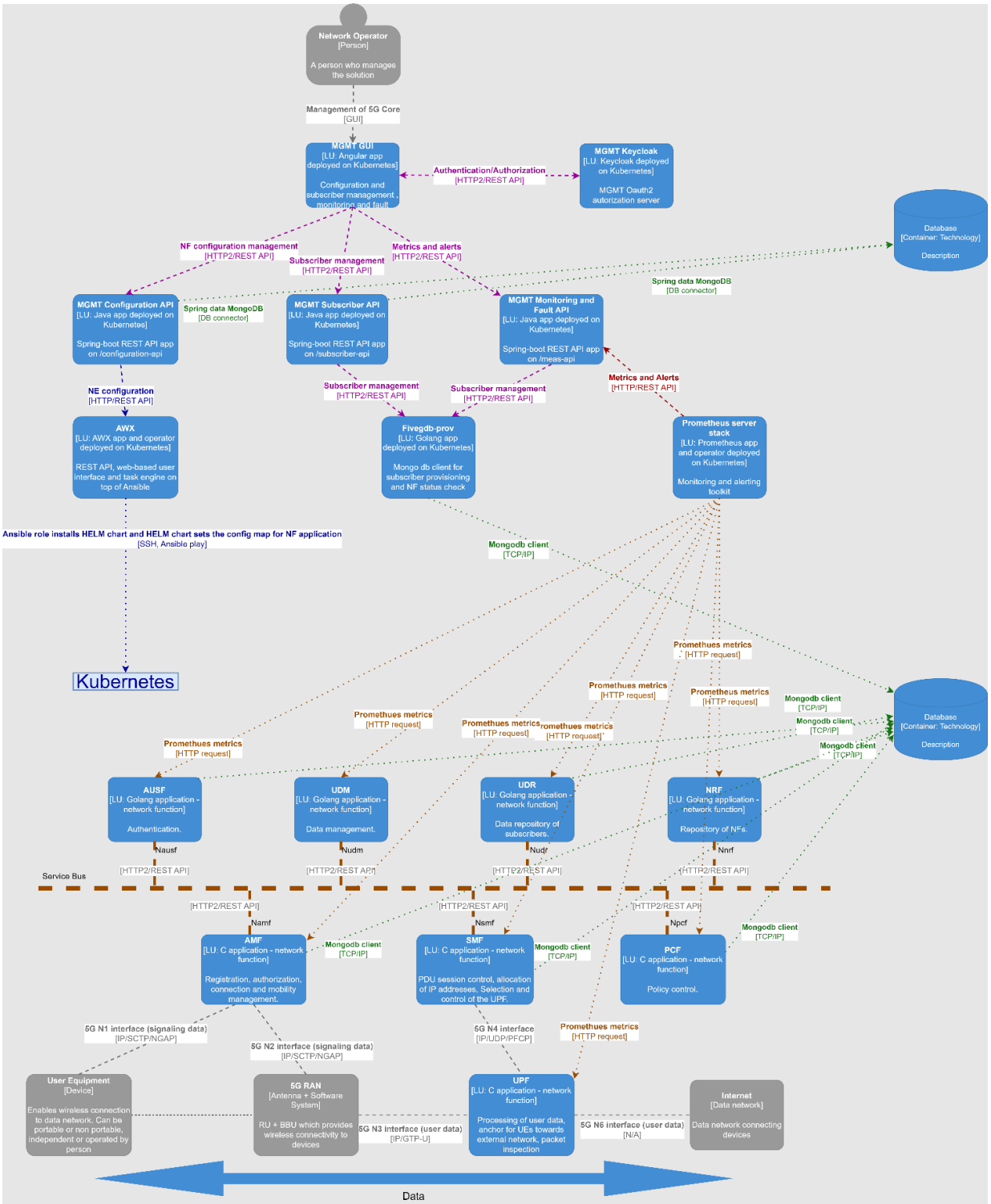


Slika 7: Združevanje segmentacije in QoS 5G omrežja

Vir: [5].

3 Kontron 5G rešitev

Kontron 5G rešitev je v prvi vrsti predvidena za mobilna privatna omrežja. Glavna naloga naše ekipe je razvoj 5G jedra, katerega postopoma razvijamo in nadgrajujemo (dopolnitve omrežnih funkcij in vpeljava novih) glede na primere uporabe in zahteve s strani končnega kupca. Od zunanjih partnerjev pridobivamo naprave RAN in UE ter medsebojno testiramo delovanje 5G jedra ter delovanje in povezljivost naprav.



Slika 8: Storitve Kontron 5G rešitve.

Kontron rešitev je sestavljena iz dveh komponent: 5G jedro ter upravitelj elementov 5G jedra. Vsaka komponenta je sestavljen iz več samostojnih storitev (slika 8).

3.1 Upravitelj elementov 5G jedra

Glavne naloge upravitelja elementov 5G jedra so:

- konfiguracija in namestitev omrežnih funkcij jedra,
- posodobitev omrežnih funkcij jedra,
- vpis in administracija naročnikov,
- prikaz stanja 5G omrežja.

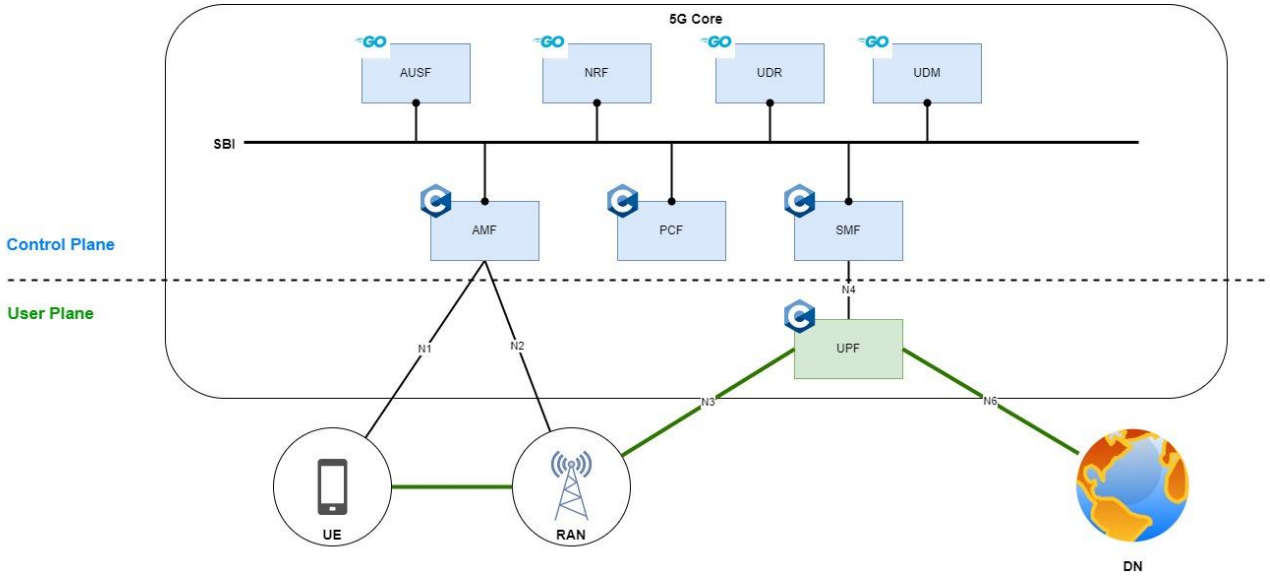
Upravitelj elementov je sestavljen iz zalednih storitev, ki so napisane v programskih jezikih Java (Spring Boot[17]) in Golang, medtem ko je za čelni del sistema[18] uporabljeno JavaScript ogrodje Angular[19].

Za upravljanje identitet in dostopov do upravitelja elementov se uporablja Keycloak[20].

Aplikacije se konfigurirajo in nameščajo s pomočjo orodja Helm[21].

3.2 5G jedro

Na sliki 9 so predstavljene omrežne funkcije Kontron 5G jedra. Delimo jih na kontrolno raven, ki skrbi za signalizacijo ter uporabniško raven, preko katere se pretaka podatkovni promet.



Slika 9: Omrežne funkcije Kontron 5G jedra.

Standard 3GPP[22] specificira omrežne funkcije kot strežnike z dobro definiranimi REST vmesniki. To nam omogoča implementacijo aplikacij v različnih programskih jezikih. Uporabljamo jezika Golang in C.
Golang omogoča hiter razvoj programske opreme z dobrimi zagotovili za dolgotrajno delovanje aplikacij. Jezik omogoča hitro prevajanje in dober nadzor nad knjižnicami in odvisnostmi. Golang ponuja številna orodja za formatiranje in testiranje kode, kar ekipi omogoča hiter razvoj aplikacij. Rezultat prevajanja so relativno majhni in učinkoviti programi, dovolj zmogljivi za večino primerov uporabe v 5G jedru.

[17] https://spring.io/projects/spring-boot
[18] čelni del sistema – izvorno "frontend", okrajšava "FE"
[19] https://angular.io/
[20] https://www.keycloak.org/
[21] https://helm.sh/
[22] https://www.3gpp.org/

Komponente 5G jedra, ki zahtevajo dodatno zmogljivost so napisane v programskem jeziku C. Primer je UPF, ki za hitro procesiranje paketov uporablja knjižnice DPDK[23].

3.3 Motivacija za prehod na oblačno zasnovano omrežje

3GPP je standardizacijska organizacija, ki piše standarde za mobilna omrežja. Pri definiciji standardov za 5G omrežje so veliko poudarka namenili vpeljavi storitveno orientirane arhitekture[24] (SBA). Ker želimo narediti 5G rešitev kompatibilno s standardom, je bila že takoj sprejeta odločitev za vpeljavo storitev z vmesniki, kot je predlagano s strani 3GPP.

Jedrne omrežne funkcije med sabo komunicirajo preko REST vmesnikov preko protokola http/2. Najdemo tudi manj razširjene omrežne protokole kot so NGAP, PFCP ter GTP-U. Protokola NGAP in GTP-U se uporabljata za komunikacijo jedra z zunanjimi elementi omrežja (UE, RAN, DN).

Jedro primarno razvijamo za privatna mobilna omrežja, zato velik poudarek namenjamo postopku namestitve in posodobitve jedra na različnih okoljih. Omenjeni postopki morajo biti čim preprostejši za končno stranko. Zagotavljati je potrebno tudi ustrezno spremljanje stanja rešitve in alarmiranje v primeru kritičnih napak.

Dodatno se od 5G jedra zahteva, da bo zagotavljalo visoko razpoložljivost omrežja, prilagodljivost in razširljivost, učinkovito porabo virov in omogočalo hitro dostavo novih rešitev. Na primer v primeru vpeljave segmentacije omrežja (poglavje 2.6), je potrebno zagotavljati dinamično spremembo topologije jedra. Dodajanje, odvzemanje ali spremembo rezin omrežja je potrebno zagotoviti med delovanjem jedra. To pomeni, da je potrebno zagotoviti namestitev ali odstranitev omrežnih funkcij in spremembo konfiguracij omrežnih funkcij brez izpada omrežja.

Glede na raznolikost rešitve in omenjene zahteve, smo se odločili za oblačno zasnovo omrežja. Storitve so pripravljene za zabojnike, ki jih orkestrira Kubernetes[25]. Kubernetes nam zagotavlja orkestracijo, razširljivost, visoko razpoložljivost, prenosljivost in lažje upravljanje konfiguracij. Poleg tega se z vpeljavo zabojnikov poenostavi namestitev rešitve na različna okolja (na strežnike pri stranki, na oblak različnih ponudnikov). Za namestitev in konfiguracijo samih komponent se uporablja orodje Helm.

Avtomatizacija z uporabo Kubernetesa nam omogoča tudi hitrejšo in lažjo vpeljavo različnih DevSecOps cevovodov za gradnjo produkta, nočno testiranje in varnostno skeniranje.

3.4 Sistemske komponente

Rešitev smo poskusili približati čim bolj generični namestitvi aplikacij na Kubernetes gručo. Aplikacije so v veliki večini spletni strežniki z REST vmesnikom, kar jih naredi optimalne uporabnike tovrstne platforme.

Nekatere aplikacije 5G jedra potrebujejo dostop do omrežnih vmesnikov, ki jih splošna Kubernetes gruča ne ponuja. Uporabo poljubnih omrežnih vmesnikov omogočimo z uporabo vtičnika Multus.

Kot podatkovno bazo uporabljamo MongoDB. Za postavitev visoko-razpoložljive baze uporabljamo Kubernetes s posebnim vtičnikom za koriščenje shrambe na gostiteljskih sistemih.

3.4.1 MongoDB

Aplikacije 5G jedra shranjujejo svoje podatke v dokumentni bazi MongoDB. Tovrstna baza optimalno teče na namenskih strežnikih izven Kubernetes gruče. Postavitev 5G jedra smo približali minimalnemu številu strežnikov, s katerim lahko zagotavljamo visoko razpoložljivost. MongoDB tako teče znotraj Kubernetes gruče na treh strežnikih in si deli vire z ostalimi aplikacijami. Z uporabo Kubernetes nastavitev "requests" in "limits" strogo nadzorujemo porabo resursov vsake aplikacije. Podatki so shranjeni na gostiteljevem podatkovnem sistemu.

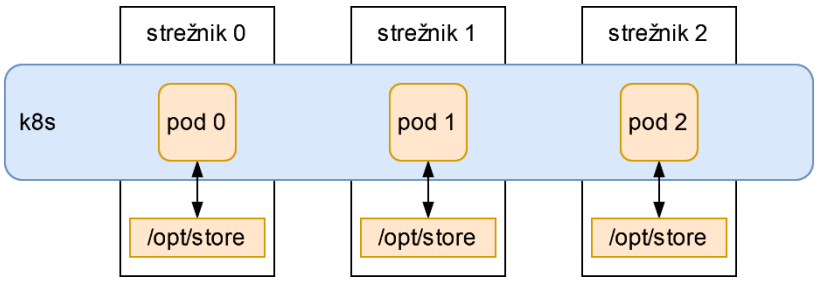
[23] https://www.dpdk.org/
[24] izvorno "Service Based Architecture", okrajšava. "SBA"
[25] https://kubernetes.io/

3.4.2 Vtičnik za shrambo

Nekatere aplikacije v Kubernetes gruči za delovanje potrebujejo obstojno shrambo ("persistent storage"). Najbolj zahtevna izmed teh aplikacij je baza (MongoDB), za katero smo potrebovali hitro in zanesljivo shrambo. Izbrali smo lokalno shrambo na gostiteljskem sistemu, ki jo implementira vtičnik local-path-provisioner[26]. Pri uporabi tovrstne shrambe moramo biti pozorni na njene omejitve:

- V privatnih omrežjih lahko predvidimo zgornjo mejo naročnikov (naprav), kar nam omogoča načrtovanje zgornje meje velikosti shrambe. Zavedamo se, da je povečava shrambe zahteven postopek, ki zahteva zaustavitev sistema.
- Gostiteljski operacijski sistem mora imeti implementirane mehanizme, s katerimi loči podatkovno shrambo od shrambe operacijskega sistema – to smo dosegli z logično particijo (LVM[27]).
- Implementiran mora biti nadzor porabe diska ter primerni alarmi.
- Aplikacije, ki zahtevajo obstojno shrambo so vezane na strežnik, na katerem je bila shramba kreirana.

Vtičnik za shrambo local-path-provisioner Kubernetes gruči ponudi t.i. shrambni razred[28] (SC). Ko aplikacija od shrambnega razreda zahteva obstojni disk[29] (PV), vtičnik na strežnikovem podatkovnem sistemu ustvari nov direktorij ter ga vpne v zabojnik.



Slika 10: vtičnik za shrambo local-path-provisioner.

3.4.3 Multus

Omrežni vtičnik Multus[30] omogoča uporabo več različnih omrežij za Kubernetes zabojnike. Multus je meta-vtičnik - to je CNI vtičnik preko katerega lahko uporabljamo več različnih CNI vtičnikov. CNI (Container Network Interface[31]) je standard, ki predpisuje API preko katerega vtičnik Kubernetes gruči ponuja omrežje za zabojnike.

Kot privzeto omrežje za zabojnike uporabljamo Cilium[32]. Poleg privzetega omrežja uporabljamo vtičnik za omrežni most ("bridge") ter vtičnik za gostiteljsko napravo ("host device").

Vtičnik "bridge" zabojniku omrežne funkcije AMF[33] pripne gostiteljev "bridge". Aplikacija AMF na ta način komunicira z zunanjim svetom preko protokola SCTP. Omeniti je treba, da nekateri Kubernetes omrežni vtičniki že podpirajo protokol SCTP, kljub temu smo se zaradi večje stabilnosti in nadzora nad prometom odločili za implementacijo preko vtičnika Multus.

[26] https://github.com/rancher/local-path-provisioner
[27] https://www.man7.org/linux/man-pages/man8/lvm.8.html
[28] Shrambni razred – izvorno "StorageClass", okrajšava "SC"
[29] Obstojni disk – izvorno "PersistentVolume", okrajšava "PV"
[30] https://github.com/k8snetworkplumbingwg/multus-cni
[31] https://www.cni.dev/
[32] https://cilium.io/
[33] AMF – Access and Mobility Function

Vtičnik "host-device" zabojniku omrežne funkcije UPF pripne vmesnika za povezavi N3 in N6, preko katerih se pretaka podatkovni promet. To omogoča aplikaciji v UPF zabojniku direkten dostop do fizičnih omrežnih vmesnikov ter optimalni izkoristek zmogljivosti omrežne kartice. Z uporabo vtičnika Multus se izognemo dodatnim obremenitvam omrežnega prometa preko Kubernetes omrežja.

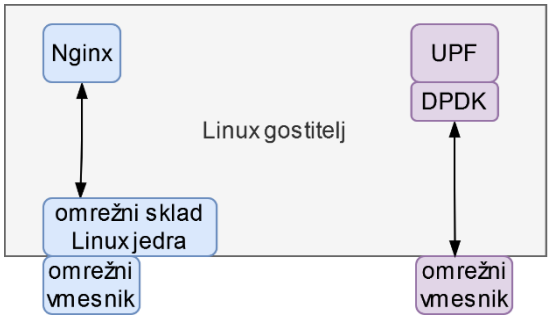
3.4.4 DPDK

Multus nam omogoča direktno uporabo omrežnih vmesnikov brez dodatne obremenitve omrežnega prometa preko Kubernetes omrežja. Omrežno kartico lahko izkoristimo še bolje.

Neposredna uporaba omrežnega vmesnika, ki ga ponuja Linux gostiteljski sistem še vedno predstavlja dodatno obremenitev in sicer procesiranje omrežnih paketov na nivoju Linux jedra. Tej dodatni obremenitvi se lahko izognemo z uporabo knjižnic DPDK.

DPDK[34] (Data Plane Development Kit ali Komplet za razvoj podatkovne ravnine) je skupek knjižnic za hitro procesiranje omrežnih paketov z ozirom na optimalno uporabo razpoložljive strojne opreme. DPDK Linux jedru odvzame nadzor nad omrežno kartico ter ga izroči aplikaciji v uporabniškem prostoru. Uporabniku omogoči razvoj lastnega omrežnega sklada, optimiziranega za specifičen primer uporabe.

V primeru 5G jedra, omrežna funkcija UPF opravlja vlogo omrežnega usmerjevalnika. UPF prejema omrežne pakete s povezav N3 ter jih glede na pravila pridobljena iz nadzorne ravni usmerja na podomrežja preko povezave N6.



Slika 11: Primerjava omrežnega sklada Linux jedra ter DPDK.

3.5 5G jedro na Kubernetes gruči

Z uporabo Kubernetes gruče smo dosegli poenoten način namestitve, posodobitev ter konfiguracije aplikacij. 5G jedro se namesti z uporabo dveh Helm chartov[35]: Helm chart za namestitev jedrnih funkcij ter Helm chart za namestitev upravitelja elementov. Namestitev predvideva Kubernetes gručo z nekaterimi vnaprej nameščenimi sistemskimi aplikacijami ter s posebnimi delovnimi vozlišči[36] za povezave N2, N3 in N6.

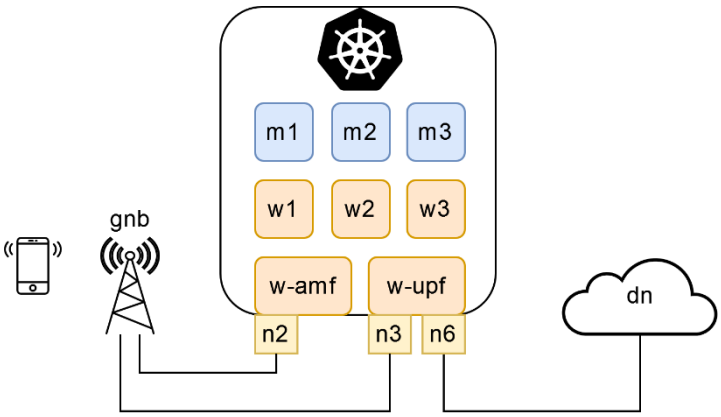
3.5.1 Omrežna arhitektura 5G jedra na Kubernetes gruči

Večina aplikacij 5G rešitve lahko teče na splošnih delovnih vozliščih Kubernetes gruče. Posebni sta omrežni aplikaciji AMF in UPF, ki zahtevata dodatne omrežne vmesnike, ter dodatno konfiguracijo omrežja izven Kubernetes gruče.

Slika 12 prikazuje primer postavitve 5G jedra na Kubernetes gruči s posebnimi delovnimi vozlišči. Kubernetes gruča je zgrajena iz treh vozlišč Kubernetes kontrolne ravni[37] (m1, m2, m3), treh splošnih delovnih vozlišč (w1, w2, w3) ter dveh posebnih delovnih vozlišč (w-amf, w-upf). Aplikacije upravitelja elementov, infrastrukturne

[34] https://www.dpdk.org/
[35] https://helm.sh/docs/topics/charts/
[36] delovno vozlišče – izvorno "worker node"
[37] Kubernetes kontrolna raven – izvorno "Kubernetes control plane"

aplikacije ter jedrne aplikacije tečejo na splošnih delovnih vozliščih. Izjema sta aplikaciji AMF in UPF, ki potrebujeta vozlišča s posebnimi omrežnimi vmesniki za komunikacijo z zunanjim svetom.



Slika 12: Omrežje 5G jedra na Kubernetes gruči.

Omrežna funkcija AMF je vezana na vozlišče w-amf, ki vsebuje omrežni vmesnik N2. Preko omrežnega vmesnika N2 teče kontrolna signalizacija med AMF in bazno postajo (gNB[38]).

Omrežna funkcija UPF teče na vozlišču w-upf, ki vsebuje omrežna vmesnika N3 ter N6. Omrežni vmesnik N3 predstavlja povezavo za podatkovni promet med UPF in bazno postajo. Omrežni vmesnik N6 predstavlja povezavo za podatkovni promet med UPF in podatkovnim omrežjem (DN).

3.5.2 Infrastrukturne aplikacije

Namestitev 5G jedra na Kubernetes gručo zahteva nekatere vnaprej nameščene infrastrukturne aplikacije:

- Nadzorni sklad Prometheus[39]
- Podatkovna baza MongoDB
- Pošiljatelj dnevnikov FluentD[40] ter zbiratelj dnevnikov Loki[41].

Kubernetes gruča mora ponujati naslednje komponente:

- Ingress controller[42]
- Shrambni razred (StorageClass)
- Omrežni vtičnik Multus

Zgoraj naštete komponente najdemo v večini produkcijskih Kubernetes gruč. Največja posebnost sta vtičnik Multus ter dodatni omrežni vmesniki za povezave N2, N3, N6, ki morajo biti na voljo preko posebnih delovnih vozlišč. Zaradi posebnih omrežnih povezav potrebujemo vsaj eno delovno vozlišče, ki nam omogoča dodajanje poljubnih omrežnih vmesnikov. Pomembno se je zavedati, da nekatere Kubernetes distribucije ne predvidevajo posegov v operacijske sisteme vozlišč. Primer je OpenShift, ki za svoja vozlišča uporablja nepremičen ("immutable") operacijski sistem CoreOS.

Vozlišča s posebnimi omrežnimi vmesniki označimo s pomočjo Kubernetes label. Te labele so vhodni podatek za Helm charte pri namestitvi jedrnih aplikacij.

[38] https://www.5g-networks.net/5g-technology/5g-terminology-the-gnb/
[39] nadzorni sklad Prometheus, izvorno "Prometheus monitoring stack", https://prometheus.io/
[40] https://www.fluentd.org/
[41] https://grafana.com/oss/loki/
[42] https://kubernetes.io/docs/concepts/services-networking/ingress-controllers/

3.5.3 Nadzor aplikacij 5G jedra

Upravitelj elementov ponuja uporabniku pregledno nadzorno ploščo, ki prikazuje stanje 5G jedra. Nadzorna plošča omogoča namestitve in posodobitve aplikacij 5G jedra. Omogoča uporabniku prijazno konfiguracijo parametrov 5G omrežja, prav tako omogoča upravljanje z uporabniki 5G omrežja.

Upravitelj elementov spremlja stanje 5G aplikacij preko metrik nadzornega sklada Prometheus. Uporabniku ponuja grafični prikaz stanja 5G aplikacij ter stanja resursov platforme. Nadzorna plošča uporabniku pomaga pravilno konfigurirati aplikacije 5G jedra, prav tako ga opozarja v primeru nepravilnega delovanja kakšne od 5G komponent. Podrobnejši pregled metrik je možen na nadzorni plošči Grafana.

3.6 DevSecOps postopki

Za interni razvoj 5G aplikacij uporabljamo številna DevSecOps orodja, za grajenje komponent, izvajanje funkcionalnih testov ter, varnostno skeniranje izdanih aplikacij.

GitLab CI/CD, Grype, Trivy, Syft, DefectDojo, Dependency Track, IXIA in UERANSIM nam pomagajo pri izvajanju varnostnih preverjanj in zagotavljanju kakovosti kode. Eno od ključnih načel DevSecOps je vključevanje varnostnih preverjanj v celotni proces CI/CD.

Varnostne smernice je treba upoštevati že v zgodnjih fazah razvojnega cikla. To pomeni, da je potrebno izvajati varnostne preglede zabojnikov, slike in kode že v fazi razvoja kode. To zahteva sodelovanje med varnostnimi strokovnjaki in razvijalci ter vključitev varnostnih orodij v avtomatizirane procese razvoja.

Cevovodi nam omogočajo izvajanje testov ranljivosti na zabojnikih, kateri nam podajo najnovejše varnostne luknje po CVE[43]. Z namenom zagotavljanja celovite varnosti upravljamo z vsemi najdenimi ranljivostmi ter jih spremljamo, dokumentiramo in odpravljamo. Za upravljanje varnostnih pomanjkljivosti in ranljivosti programske opreme uporabljamo orodje Defect Dojo[44].

S pomočjo specializiranih orodij izvajamo statično analizo kode. Cevovod vključuje gradnjo SBOM[45], ki omogoča natančno sledenje in upravljanje vseh komponent našega sistema. S tem se zagotavlja transparentnost in sledljivost vseh elementov.

Za preverjanje delovanja in ustreznosti rešitve, nočni cevovodi redno izvajajo nočne regresijske teste. Poln nabor testov se izvede tudi pred vsako objavo novega programskega paketa. S tem zagotavljamo, da je vsak korak preizkušen, preden se nadaljuje z objavo programskega paketa in s postopki verificiranja. Napake ugotovljene med avtomatičnimi testi se rešuje prioritetno.

CI/CD cevovod izvaja avtomatsko namestitev 5G rešitve z uporabo orodja Helm. To orodje nam zagotavlja ponovljivost namestitve ter sledenje celotni namestitveni konfiguraciji v sistemu za verzioniranje.

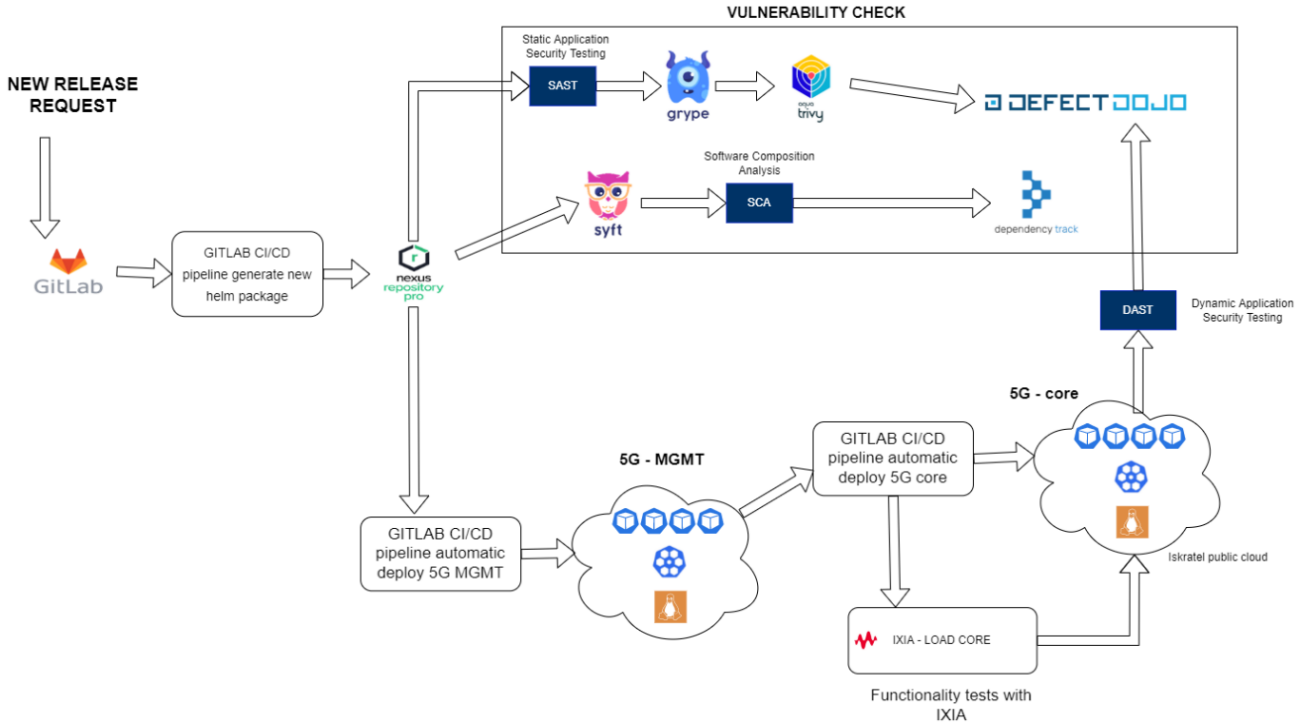
Vsebina programskega paketa je definirana v konfiguracijski datoteki, kjer so shranjene verzije posameznih komponent (verzije Helm chartov ter verzije Docker slik). V datoteki se natančno opredeli vse potrebne komponente, njihove verzije ter druge pomembne informacije, ki so bistvene za postopek gradnje paketa. S tem se zagotovi doslednost in pravilno usklajenost med različnimi cevovodi ter omogoča učinkovito izvajanje nadaljnjih korakov. Pred objavo paketa se izvede vsa prej omenjena preverjanja (preverjanje ranljivosti, statična analiza kode, testi).

Cevovode se še vedno nadgrajuje in dopolnjuje z namenom avtomatizacije in optimizacije vseh omenjenih korakov ter vpeljavo novih.

[43] CVE – izvorno "Common Vulnerabilities and Exposures", https://www.cve.org/
[44] https://www.defectdojo.org/
[45] seznam sestavnih delov – izvorno "Software Bill of Materials", okrajšava "SBOM"

3.6.1 Cevovod za gradnjo novega paketa

Na sliki 13 je prikazan cevovod, ki se sproži ob zahtevi za gradnjo nove paketne izdaje 5G rešitve. Osnovni cevovod vključuje tudi klice dodatnih cevovodov za specifične naloge.



Slika 13: Prikaz cevovoda ob gradnji nove paketne izdaje.

Glavni koraki cevovoda, ki se izvedejo ob zahtevi za nov paket 5G rešitve:

- Gradnja Docker slik in Helm chartov, shranjevanje rezultatov gradnje v Nexus[46],
- Cevovod za avtomatsko postavitev upravitelja jedra,
- Cevovod za avtomatsko postavitev 5G jedra,
- Avtomatski testi za 5G jedro z orodjem IXIA[47],
- Dinamična analiza kode ter odlaganje poročil na Defect Dojo,
- Generiranje SBOM na vseh zabojnikih in odlaganje poročil na Dependency Track[48],
- Statična analiza kode, ranljivostni testi zabojnikov z orodji Grype in Trivy, odlaganje poročil na Defect Dojo.

[46] Repozitorij, https://www.sonatype.com/products/sonatype-nexus-repository
[47] https://support.ixiacom.com/

4 Zaključek

5G zagotavlja številne prednosti in izboljšave v primerjavi s prejšnjimi generacijami mobilnih omrežij. Za zagotavljanje večjih hitrosti prenosa, nižjih zakasnitev, večje omrežne zmogljivosti in večje varnosti, igra ključno vlogo jedro 5G omrežja.

V članku je predstavljeno jedro 5G omrežja kot kompleksna rešitev, ki mora omogočati razširljivost, zanesljivost in prilagodljivost za namestitev pri različnih strankah. Omenjene zahteve se rešuje s postavitvijo zabojnikov 5G rešitve v oblačno zasnovano omrežje. Za zagotavljanje večjih hitrosti in manjših zakasnitev uporabniškega prometa se uporabi specifične sistemske komponente (Multus, DPDK).

Zaradi uporabe različnih tehnologij in programskih jezikov je potrebno veliko pozornosti in časa nameniti DevSecOps postopkom. Začeten vložek v gradnjo cevovodov se kmalu povrne v obliki stabilne in zanesljive rešitve. Poleg tega je potrebno nameniti velik poudarek tudi varnostnemu preverjanju rešitve že od zgodnje faze razvoja kode in vse do končne postavitve rešitve.

Ob razvoju 5G rešitve kot oblačno zasnovanega omrežja smo prišli do ugotovitve, da je potrebno za samo obvladovanje namestitev in posodobitve rešitve ter kvalitetnih DevSecOps postopkov nameniti veliko časa in vložka. Poleg tega je ključno tesno sodelovanje med razvojno in DevSecOps ekipo.

Literatura

[1] https://www.3gpp.org/technologies/5g-system-overview, 3rd Generation Partnership Project (3GPP), obiskano 28.6.2023

[2] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; System architecture for the 5G System (5GS); Stage 2 (Release 16)

[3] https://blog.se.com/telecommunications/2022/07/28/the-evolution-of-5g-and-the-backup-power-it-requires/, obiskano 4.7.2023

[4] https://www.3gpp.org/news-events/3gpp-news/sys-architecture, obiskano 10.7.2023

[5] https://devopedia.org/5g-quality-of-service, obiskano 10.7.2023

[6] https://www.redhat.com/en/topics/cloud-native-apps/vnf-and-cnf-whats-the-difference, obiskano 13.7.2023

[7] https://www.analog.com/en/analog-dialogue/articles/massive-mimo-and-beamforming-the-signal-processing-behind-the-5g-buzzwords.html, obiskano 4.7.2023