UNIVERZITET “UNION”

RAČUNARSKI FAKULTET

Knez Mihailova 6/VI
11000 BEOGRAD

Arhitektura platforme za analitiku podataka u oblaku

DIPLOMSKI RAD

Student: **Branko Fulurija RN 3/16** Mentor: **dr Dušan Vujošević**

Beograd, septembar 2020.

Apstrakt

Podaci su u današnjem poslovnom i tehnološkom svetu neophodan i neizostavan deo svake uspešne organizacije. Tehnologije i alati za rad sa velikim skupovima podataka doživeli su veliki napredak i inovacije u protekloj deceniji i na taj način omogućili da analitika postane dostupna većem broju organizacija različitih veličina. Upotreba analitike sada više nije ograničena samo na tehnološke gigante sa dobrim finansijskim stanjem. Obrada i analiza podataka je postala rasprostranjena i prema sprovedenim istraživanjima, čak oko 60% organizacija koriste analitiku u nekom obliku [8]. U današnjem svetu, upravljanje kompanijom bez upotrebe analitike se može uporediti sa vožnjom automobila u nepoznatoj zemlji bez pomoći mape ili GPS-a.

U ovom radu su razmatrane najveće koristi koje organizacije danas mogu da dobiju adekvatnom upotrebom obrade i analize svojih podataka. Naravno, da bi kompanije zapravo došle od sirovih i neobrađenih podataka do zlata vrednih uvida i novih informacija, neophodna je odgovarajuća infrastruktura i implementiran sistem za analitiku. Na početku rada se razmatraju tradicionalna rešenja za ovakve sisteme i analiziraju se potencijalni problemi na koje ovi sistemi nailaze, prvenstveno zbog ekspanzije koncepta velikih podataka. Nakon tradicionalnih rešenja, analizira se arhitektura sistema za analitiku gde je centralna komponenta skladište podataka (*data warehouse*), mesto koje će da sadrži sve podatke bitne za jednu organizaciju. Ovaj pristup, iako široko rasprostranjen u današnjoj industriji, nije idealan za sve primene, a glavni krivac za ovo je razvoj novih tehnologija poput interneta od stvari (*internet of things, IoT*), mašinskog učenja i veštačke inteligencije, koje generišu i koriste ogromne količine veoma raznovrsnih podataka velikom brzinom.

Cilj ovog rada je predstavljanje koncepta platforme za analitiku podataka, modernog sistema u oblaku sa slojevitom arhitekturom, gde je svaki sloj zadužen za jednu specifičnu funkciju. Uvodi se pojam jezera podataka (*data lake*) i navode se prednosti integracije i koegzistencije jezera i skladišta podataka u okviru jedne platforme. Pojedinačni slojevi arhitekture platforme se detaljno istražuju i navode se konkretni primeri modernih servisa u oblaku koji se mogu koristiti za implementaciju datog sloja. Rezultat rada je predlog jednog pristupa izrade arhitekture moderne platforme za analitiku, spremne da se izbori sa raznovrsnim problemima u današnjem svetu velikih podataka, društvenih mreža, interneta od stvari i veštačke inteligencije.

Ključne reči: veliki podaci, analitika podataka, oblak, skladište podataka, jezero podataka, slojevita arhitektura, platforma, ETL, ELT, modelovanje podataka, poslovna inteligencija, *Google Cloud*, veštačka inteligencija, strim.

Sadržaj

1. Uvod

Danas živimo u svetu velikih podataka. Svakog dana se generiše ogromna količina informacija, a mogućnost i sposobnost čuvanja, obrade i analize podataka kao i izvlačenje novih informacija jeste obaveza za sve kompanije koje žele da unaprede i razvijaju svoje poslovanje.

Potencijal velikih podataka nastavlja da raste. Kako bi organizacije izvukle maksimum iz njih, one moraju da uračunaju analitiku u svoju strategiju i viziju da bi bile u mogućnosti da donose bolje i brže odluke. Količina generisanih podataka nastavlja da se udvostručuje svake tri godine [10]. Uzrok ovome je razvoj digitalnih platformi, bežičnih senzora, aplikacija virtuelne realnosti i milijarde mobilnih telefona koji generišu velike količine podataka u kratkom vremenskom periodu. Postoje istraživanja koja pokazuju da je 90% svih svetskih podataka generisano u poslednje dve godine [10]. Ova istraživanja nas upućuju na činjenicu da je sada više nego ikad neophodno iskoristiti zlatni potencijal koji je sakriven u gomili neobrađenih podataka. Jedan od glavnih proizvođača podataka jesu povezani (pametni) uređaji. Procene su da danas postoji oko 30 milijardi takvih uređaja, što je oko 5 pametnih uređaja po svakom čoveku na Zemlji [11]. Neki od ovih uređaja generišu desetine ili stotine događaja svake sekunde.

Tehnološki napredak nas je doveo u položaj da možemo da se izborimo sa ovom količinom informacija. Kapacitet uređaja za skladištenje podataka je znatno porastao, dok je nasuprot tome njihova cena drastično opala [13]. Inženjeri koji se bave obradom i analizom podataka, danas imaju, nekada nezamislivu količinu računarske snage na raspolaganju i dolaze do novih sofisticiranijih algoritama. Kompanije koriste obradu i analizu podataka ne samo da unaprede svaki aspekt svog poslovanja i operacija, već da ustanove i potpuno nove biznis modele. Revolucija i razvoj računarstva u oblaku omogućila je malim i srednjim kompanijama da otključaju vrednosti podataka koje skladište, iako su takve prednosti ranije bile rezervisani samo za gigante tehnološke industrije.

1.1 Analitika podataka

Analitika podataka (*data analytics*) je proučavanje sirovog i neobrađenog skupa podataka služeći se pritom statističkim i analitičkim alatima za pretvaranje tih podataka u smislene informacije i izvršavanje jednog ili više zadataka. Analitici se pristupa da bi se iz ogromne količine podataka raščlanjivanjem i pronalaženjem pravilnosti došlo do pojednostavljenih rezultata. Glavni uslov za kvalitetnu analitiku jeste prikupljanje dovoljne količine podataka iz odgovarajućih izvora. U procesu prikupljanja podatka, određeni podskup se možda ne može koristiti i takve podatke je neophodno odstraniti iz sistema kako ne bi došlo do usporavanja analitike ili njenog usmeravanja u pogrešnom pravcu. Mnoge tehnike i procesi uključeni u analitiku podataka su automatizovani u mehaničke procese i algoritme koji neobrađene podatke pretvaraju u strukturirane i obrađene skupove spremne za upotrebu.

1.1.1 Analitički proces

Analitika podataka je širok pojam koji obuhvata veliki broj različitih tipova analize podataka. Bilo koja informacija može biti podvrgnuta analitičkim tehnikama da bi se došlo do novih uviđanja i shvatanja.

Kompanije koje se bave proizvodnjom često beleže radno vreme, vreme u kvaru i količinu posla na čekanju za razne mašine koje poseduju i nakon toga analiziraju taj skup podataka kako bi bolje planirali radno opterećenje tako da mašine rade blizu svog maksimalnog kapaciteta, i da iskorišćenost bude optimizovana. Osim detekcije problematičnog dela i uskog grla u proizvodnji, analitika podataka može da ima druge koristi za organizacije. Gejming kompanije koriste analitiku da utvrde redosled nagrađivanja korisnika kako bi obezbedili da veliki broj korisnika aktivno igra njihove igrice. Kompanije koje se bave plasiranjem sadržaja koriste analitiku da bi zadržali korisnike na svojim platformama.

Proces koji prati analitika podataka uključuje nekoliko različitih koraka [12]:

1. **Definisanje pravih pitanja** - U organizacionoj ili biznis analizi, neophodno je početi sa pravim pitanjima. Pitanja treba da su jasna, koncizna i da omoguće merljivost. Pitanja je potrebno osmisliti tako da ili kvalifikuju ili diskvalifikuju potencijalna rešenja za dati problem.
2. **Postavka jasnih prioriteta merenja** - Ovaj korak se može podeliti u dva dela:
    - Određivanje šta će se meriti - Neophodno je razmotriti koje metrike su nam potrebne da bi dali odgovor na naša ključna pitanja koja smo ranije definisali. Ovakav skup metrika se naziva glavnim indikatorima performansi (*key performance indicators, KPI*)
    - Određivanje kako će se meriti - Potrebno je razmisliti o tome kako ćemo meriti podatke koji će se sakupljati. Primer za ovo može biti odluka koji ćemo vremenski interval da koristimo (dnevni, mesečni, kvartalni i dr.) ili odluka u kojoj ćemo valuti računati prihode (evro, dolar i dr.)
3. **Prikupljanje podataka** - Kada imamo jasno definisana pitanja i prioritete za metrike, vreme je za razvijanje strategije za prikupljanje podataka. U ovom koraku je neophodno da počnemo sa sakupljanjem pravih podataka iz izvora koji će nam pomoći da damo odgovor na naša ključna pitanja i merimo već definisane indikatore performansi.
4. **Analiza podataka** - U ovom koraku je neophodno primeniti razne manje i više sofisticirane softverske alate i algoritme kako bi izvršili adekvatnu analizu. Dobra polazna tačka jeste manipulacija podacima na različite načine, primenom deskriptivne statistike. Prilikom manipulacije podacima, može se desiti da zapravo imamo sve podatke koji su nam neophodni, ali češći slučaj jeste da u tom trenutku shvatimo da nam je potrebna revizija početnog pitanja ili sakupljanje dodatnih podataka. U svakom slučaju, početna analiza trendova, korelacija, varijacija i izuzetaka pomaže da se fokusiramo na davanje tačnijih i boljih odgovora na ključna pitanja.
5. **Interpretacija rezultata** - Nakon analize podataka i potencijalno izvođenja daljeg istraživanja, konačno je vreme za interpretaciju rezultata. Prilikom interpretacije analize, neophodno je biti svestan da nikad nećemo moći da dokažemo da je hipoteza zapravo tačna, jedino ćemo biti u mogućnosti da odbacimo hipotezu kao netačnu. Prilikom interpretacije rezultata, treba obratiti pažnju na sledeća pitanja:

- Da li rezultati daju odgovor na početna pitanja i na koji način?
- Da li rezultati pomažu u odbrani od raznih prigovora i na koji način?
- Da li postoje određena ograničenja za naše zaključke i koje uglove potencijalno nismo razmotrili?

1.1.2 Tipovi analitike

Postoje četiri osnovne vrste analitike podataka. U nastavku će biti izložene od jednostavnijih do sofisticiranijih. Što je tip analitike složeniji to je i kompleksnija analiza, ali i znatno veća vrednost koju donose rezultati analize.

Osnovni tipovi analitike podataka [9]:

1. **Deskriptivna analitika** - Daje odgovor na pitanje šta se dogodilo i smatra se za najjednostavniju formu analitike. Količina velikih podataka za koju organizacije danas koriste uključuje razbijanje velikog skupa podataka u razumljive celine. Svrha ovog tipa analitike jeste sumiranje pronalazaka i razumevanje šta se zapravo dogodilo u prošlosti. Glavne tehnike koje se koriste su: ad-hok izveštavanje, rudarenje podataka, agregiranje podataka i sumarna statistika.
2. **Dijagnostička analitika** - Daje odgovor na pitanje zašto se nešto dogodilo. Dijagnostička analitika pokušava bolje da razume glavni uzrok određenih događaja. Korisna je u određivanju koji su faktori i događaji doprineli ishodu. Uglavnom koristi verovatnoće, značajnost i raspodelu rezultata za analizu. Glavne tehnike koje se koriste su: analiza glavnih komponenti (*principal component analysis*), analiza osetljivosti i regresiona analiza. Algoritmi mašinskog učenja poput klasifikacije i regresije takođe spadaju u ovaj tip analitika.
3. **Prediktivna analitika** - Daje odgovor na pitanje šta bi se moglo desiti ako su određeni uslovi zadovoljeni. Prediktivna analitika se koristi za određivanje budućih ishoda, ali bitno je napomenuti da ona zapravo ne može sa sigurnošću reći da li će se neki događaj odigrati u budućnosti, već samo predviđa koje su verovatnoće da će se taj događaj zapravo dogoditi. Prediktivni modeli se grade na osnovu preliminarnih faza deskriptivne analitike. Glavne tehnike koje se koriste su: kvantitativna analiza, prediktivno modelovanje, algoritmi mašinskog učenja.
4. **Preskriptivna analitika** - Daje odgovor na pitanje koje akcije su najbolje u zavisnosti od željenih ishoda. Osnova ove faze jeste prediktivna analitika, ali ona prevazilazi sve tri gore pomenuta tipa tako što predlaže buduća rešenja i akcije. Može da predloži sve povoljne ishode prema određenom toku akcije i takođe predlaže razne tokove akcija da bi se došlo do određenog ishoda. Ona zapravo koristi snažan sistem povratnih informacija koji stalno uči i ažurira odnos između akcije i ishoda. Glavne tehnike koje se koriste su: simulaciona analiza, sistemi za preporučivanje, veštačka inteligencija, neuralne mreže.

1.1.3 Značaj analitike za razvoj biznisa

Gotovo sve najveće svetske organizacije, lideri u svojim oblastima, za sebe sa ponosom govore da su kompanije koje se vode podacima (*data-driven companies*). Jednostavno rečeno, svako preduzeće koje donosi poslovne odluke na osnovu činjenica, a ne na osnovu instinkta,

subjektivnog mišljenja i emocija, jeste kompanija koja se zasniva na podacima. U organizaciji zasnovanoj na podacima, ne samo da više rukovodstvo donosi odluke zasnovane na podacima, već se sve odluke na svim nivoima donose na osnovu činjenica.

Neke od najbitnijih prednosti analitika za razvoj biznisa su [13]:

1. **Bolje targetiranje** - Uz pomoć analize podataka, moguće je odrediti koji oblici oglašavanja zapravo dolaze do kupaca efektivno i imaju takav uticaj da će podstaći korisnika da kupi dati proizvod. Podaci nam omogućavaju da razumemo koji oblici oglašavanje i reklamiranja naših proizvoda imaju najveći uticaj na ciljno tržište i u kojoj meri možemo adaptirati to oglašavanje.
2. **Poznavanje ciljnih kupaca** - Jedna od najvećih koristi analitika podataka jeste provera performansi kompanijskih proizvoda na tržištu. Jednom kada razumemo koji proizvodi su pogodni za koje klijente, onda možemo da odredimo na koje oblasti će se kompanija fokusirati i na koje korisnike. Sve ove informacije su nam korisne prilikom određivanja cene i načina reklamiranja, kao i oblasti u kojoj će se organizacija specijalizovati.
3. **Inovacije** - Analiza podataka nam može dati grubu predstavu budućih trendova korisničkog ponašanja. Ovakva saznanja nam omogućavaju da gradimo nove proizvode pune futurističkih inovacija. Na ovaj način se grade proizvodi i servisi koji će potencijalni biti na vrhu svoje industrije i ostvaruje se značajna prednost nad konkurencijom.
4. **Smanjenje troškova poslovanja** - Veoma značajna prednost koju dobijamo adekvatnom primenom analitike jeste efikasno i efektivno vođenje poslovanja. Organizacije sa dobrim sistemom za analitiku su u stanju da odrede koji sektori koriste nepotrebnu količinu finansija, kao i oblasti koje zahtevaju veća ulaganja. Kroz posmatranje jasnih i preciznih činjenica, kompanija je u stanju da smanji operativne i produkcione troškove. Analiza podataka čini svaku akciju preciznom i neophodnom, dok istovremeno eliminiše akcije koje ne donose dodatnu vrednost organizaciji.
5. **Pomoć pri rešavanju problema** - Svaki problem na koji kompanija naiđe u svom poslovanju može dovesti do određenog zaustavljanja standardnog poslovanja i prouzrokovati velike troškove za kompaniju. Analiza podataka pomaže organizaciji prilikom donošenja informisanih odluka u vezi poslovanjem i na taj način sprečava pojave gubitaka. Analizirani podaci mogu se koristiti za detekciju problematičnih poslovnih i tehičkih sistema.
6. **Brže i bolje donošenje odluka** - Dobro implementirani sistemi za analitiku, pomažu višem rukovodstvu kao i svakom članu organizacije da zapravo razume prošlo, trenutno i potencijalno buduće stanje kompanije, proizvoda i kupaca. Ovakav pregled poslovanja od 360 stepeni pomaže organizaciji da donosi odluke na osnovu stvarnih činjenjica.

1.2 Tradicionalni pristup analitici

Činjenica je da svakoj organizaciji neophodna analitika, bez obzira da li rukovodstvo to shvata ili ne [1]. Oduvek je postojala potreba da se mere bitne biznis metrike i da se donose odluke na osnovu činjenica. Pitanja poput “Koliko proizvoda smo prodali prošlog meseca?” i “Koji je najbrži način za dostavu paketa od A do B?” su se razvila u morednija pitanja oblika “Koliko korisnika sajta je platilo premijum pretplatu?” i “Šta nam podaci sa IoT senzora govore o ponašanju korisnika?” [1]. Upravljanje kompanijom bez analitike se može uporediti sa vožnjom automobila u nepoznatoj zemlji bez pomoći GPS-a [2].

Pre nego što su računari postali sveprisutni, kompanije su se oslanjale na poslovodne knjige, inventare, sopstvenu intuiciju i ostale tehnike ručnog praćenja i analiziranja poslovnih metrika. Tokom kasnih 1980-ih godina pojavio se koncept skladišta podataka [2]. Skladište podataka (*data warehouse*) predstavlja repozitorijum podataka ujedinjenih iz različitih izvora. U to vreme, glavna svrha skladišta je bilo generisanje statičkih izveštaja. Naoružane ovom inovacijom, kompanije su bile u mogućnosti da donošenje odluka zasnovano na emocijama, instinktima i intuiciji zamene sa informisanim donošenjem odluka na osnovu činjeničnih podataka [1].

U nastavku ćemo pogledati neke od tradicionalnih i jednostavnijih mogućnosti koje su kompanije implementirale i koristile za analitiku podataka. Tehnološka kompleksnost sistema za analitiku najviše zavisi od veličine kompanije i njenih trenutnih potreba za praćenjem metrika. Rešenja ćemo razmatrati u redosledu rastuće složenosti.

Većina sistema za analitiku ima određene zajedničke zahteve. Tri zajedničke stvari koje skoro svaki sistem za analitiku treba da poseduje su [2]:

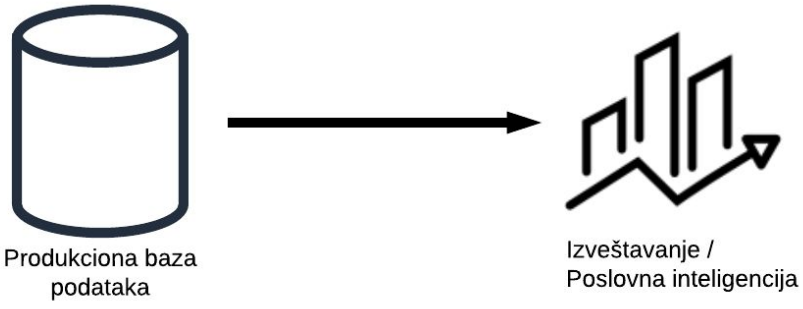
1. Potrebno je učitati podatke u centralni repozitorijum, a to je često skladište podataka.
2. Potrebno je transformisati i modelovati podatke da budu spremni za brze upite u skladišta.
3. Potrebno je dostaviti podatke do donosilaca odluka, koji su zapravo krajnji korisnici.

1.2.1 Upiti nad produkcionom bazom

Kompanije koje se nalaze u početnim fazama razvoja proizvoda ili poslovanja potencijalno mogu da preskoče neke od gore navedenih koraka u zavisnosti od trenutnog stanja kompanije i proizvoda. Za ove organizacije moguće su sledeće olakšice prilikom razvoja sistema za analitiku [2]:

- Ukoliko podaci dolaze iz samo jednog izvora (najčešće produkciona baza podataka), onda može da se preskoči korak učitavanja podataka u centralni repozitorijum.
- Ukoliko trenutni proizvod nema veliki broj korisnika tada izvršavanje analitičkih upita direktno na produkcionoj bazi neće napraviti preterano opterećenje na aplikaciju.
- Ako su neobrađeni podaci dovoljno jednostavni da se direktno koriste i interpretiraju ili ako su poslovni izveštaji jednostavni, pa ne zahtevaju kompleksne transformacije podataka, tada se može preskočiti korak transformacije i modelovanja.

U ovom slučaju platforma za analitiku može biti krajnje jednostavna: integrišemo alat za poslovnu inteligenciju (*business intelligence*) sa produkcionom bazom naše aplikacije.



Slika 1. Jednostavna postavka platforme za analitiku direktno na produkcionoj bazi [2]

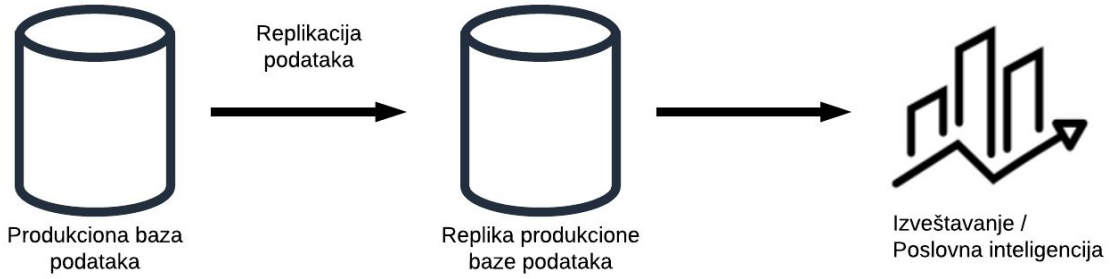
Na ovaj način će se pri upotrebi alata za poslovnu inteligenciju kreirati SQL (*structured query language*) upiti koji će se izvršavati direktno na produkcionoj bazi aplikacije. Kao rezultat ove postavke, podaci u grafikonima će se osvežavati u realnom vremenu kao posledica novih podataka u aplikacionoj bazi.

1.2.2 Upiti nad replikom baze

Postavka platforme za analitiku u kojoj se upiti direktno izvršavaju na produkcionoj bazi aplikacije jeste jednostavna za implementaciju i pogodna za manje organizacije, ali ova jednostavnost sa sobom nosi veliki broj potencijalnih mana.

Najveći rizik u ovakvom sistemu jesu performanse izveštavanja, ali još bitnije su performanse aplikacije. Osim produkcionih opterećenja sa kojima se nosi aplikaciona baza, sada postaje zadužena i za opterećenje koje dolazi od sistema za analitiku i to može da pogorša iskustvo krajnjih korisnika.

Jedno od rešenja za ovaj problem sa performansama jeste da, umesto što direktno nakačimo alat za poslovnu inteligenciju na produkcionu bazu, možemo da napravimo repliku produkcione baze i da na nju nakačimo sisteme za analitiku. Replika baze podataka predstavlja zasebnu i izolovanu instancu baze koja sadrži podatke replicirane iz glavne produkcione baze. Kada govorimo o standardnim relacionim bazama podataka, tada replike u većini slučajeva služe samo za čitanje. Svaka izmena na produkcionoj bazi će se u jednom trenutku nakon intervala zadrške, replicirati na instancu replike.



Slika 2. Postavka platforme za analitiku sa replikom produkcione baze podataka [2]

Jednostavnije postavke replike uključuju mehanizam u kojem će se sa vremena na vreme (npr. jednom dnevno) dampovati (*dump*) podaci iz produkcione baze i nakon toga učitati u instancu replike. Međutim, danas postoje napredniji sistemi i servisi za rad sa relacionim bazama koji omogućavaju programerima da relativno lako uspostave sistem replikacije podataka sa produkcione instance na repliku.

Ovakva postavka sistema uklanja prethodno opterećenje sa produkcione baze i sada će instanca replike biti zadužena za opsluživanje analitičkih zadataka i upita. Podaci koji se nalaze u replici su relativno sveži i zavise od konfigurisanog intervala repliciranja podataka između baza podataka.

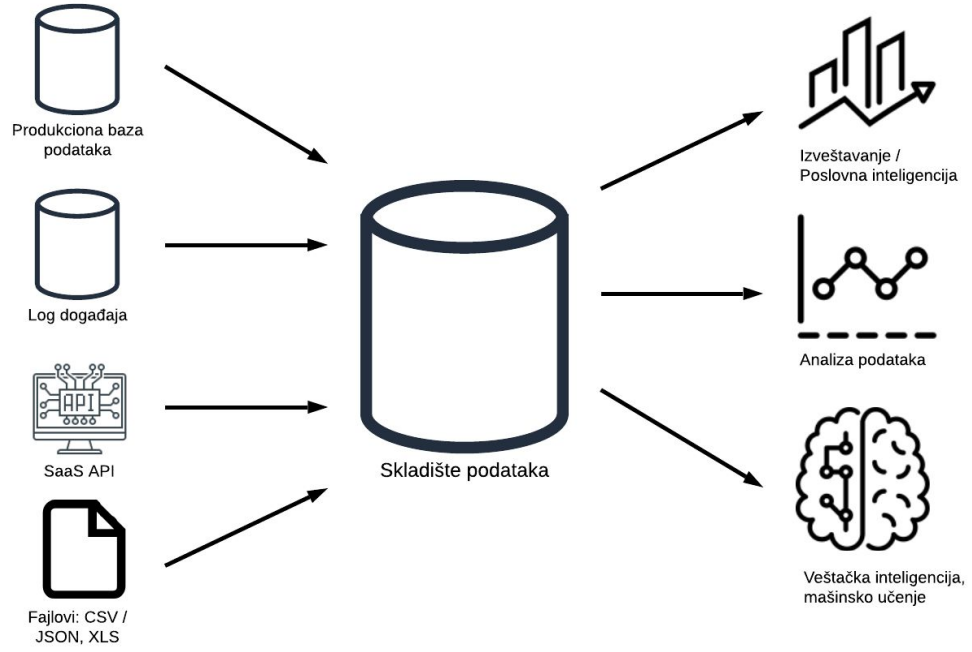
1.3.3 Uvođenje skladišta podataka

Kompanije koje imaju veoma jednostavne zahteve za analitiku mogu da se zadovolje sa gore navedenim rešenjima, ali u današnjem svetu gde se tehnoglogija veoma brzo menja, mnogim organizacijama je neophodno sveobuhvatnije rešenje.

Činjenice koje primoravaju organizacije da koriste skladište podataka:

- Kompanija koristi NoSQL aplikacionu bazu podataka - Dosta kompanija se opredeljuje za neku od NoSQL podataka za podršku svojim aplikacijama, gde su MongoDB i *Cassandra* neki od popularnih izbora. Ovaj tip baze podataka nije najpogodniji za analitičke upite iz više razloga. Prvi i veoma bitan razlog jeste činjenica da je broj alata za izveštavanje koji rade sa NoSQL bazom mnogo manji od istih alata koji rade sa relacionim bazama podataka. Drugi razlog je što NoSQL baze podataka nisu optimizovane za analitičke upite koji često uključuju operacije agergacije ili spajanja više različitih skupova.
- OLAP protiv OLTP - Standardne relacione baze podataka koji su široko zastupljene u privredi (npr. MySQL i PostgreSQL) su takozvane OLTP (*online transactional proccessing*) baze podataka. Ovaj tip baze podataka je orjentisan na upite transakcionog tipa. OLTP omogućava veoma efikasne operacije ubacivanja, izmene i brisanja manjeg broja redova neke tabele, kao i efikasno čitanje manjeg broja redova koji su filtrirani po određenim atributima. Nasuprot ovome, OLAP (*online analytical processing*) baze podataka su napravljene isključivo za svrhu podrške analitici. OLAP baze su veoma dobre u ubacivanju velikog broja redova (*batch insert*) istovremeno, ali imaju loše performanse za učestale operacije izmene pojedinačnih redova. Najveća prednost OLAP baza u odnosu na OLTP baze jeste njihova arhitektura koja im omogućava da izvršavaju upite koje obrađuju, kombinuju, filtriraju ili agregiraju veliki broj redova i kolona kako bi za relativno kratko vreme izvršili kompleksne analitičke upite. Skladište podataka je OLAP baza podataka namenjena za analitiku.
- U današnjem vremenu većina kompanija nema samo jednu centralnu aplikacionu bazu podataka, već ih ima nekoliko različitih. Primer za ovo mogu biti pojedinačni servisi u danas popularnoj mikroservisnoj arhitekturi gde svaki servis ima svoju bazu podataka. Razvoj SaaS (*software as a service*) proizvoda koje organizacije koriste kao pomoć u poslovanju takođe je povećao raznovrsnost i količinu različitih izvora podataka u okviru jedne organizacije. Svi ovi izvori su najčešće dosta različiti i potrebno ih je konsolidovati i organizovati na jedno mesto kako bi se mogli međusobno kombinovati i iskoristiti za spoznaju novih činjenica.

Skladište podataka je centralna analitička baza podataka u najvećem broju organizacija. Skladište nam služi kao mesto u kojem ćemo konsolidovati podatke iz različitih izvora i transformisati ih na takav način da ćemo alatima za poslovnu inteligenciju ili internim korisnicima omogućiti veoma efikasne upite.



Slika 3. Postavka platforme za analitiku sa skadištem podataka [2]

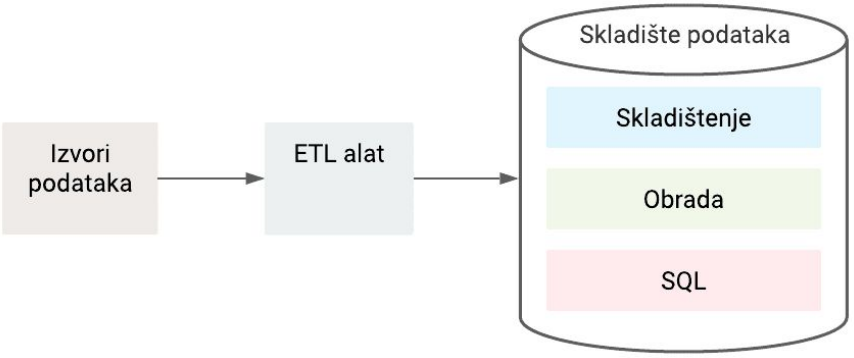
Osnovne funkcionalnosti koje svi današnji sistemi za analitiku treba da sadrže [2]:

- Podaci moraju da se prikupe, konsoliduju i sačuvaju u centralnom skladištu.
- Podaci se moraju obraditi: transformisati, agregirati i modelovati za efikasnu upotrebu u okviru skladišta.
- Podaci se moraju prezentovati: vizuelizacija, ekstraktovanje podataka i slanje podataka do servisa koji ih koriste.

1.3 Veliki podaci - veliki problem

Tradicionalna skladišta podataka imaju svoje limitacije i ograničenja. Razmotrićemo koji su to problemi na koje nailaze ova tradicionalne arhitekture kada se suoče sa velikom količinom raznovrsnih podataka koja se u današnjem vremenu proizvode ekstremnom brzinom.

Arhitektura tradicionalnih rešenja za skladištenje podataka je najčešće uključivala dve povezane komponente: skladište i obradu. Skladišta su se obično kombinovala sa nekim ETL (*extract transform load*) alatima koji su obrađivali i dostavljalu podatku u okviru tabela u skladištu podataka po unapred definisanom rasporedu. Sve komponente jednog skladišta podataka su se najčešće izvršavale na zajedničkom računaru, što je značilo da su kapaciteti za skladište bili usko vezani sa kapacitetima obrade i da se ne mogu odvojeno skalirati. Još jedno ograničenje koje se nameće jeste to što je obrada podataka ograničena samo na upotrebu jezika koji koristi to specifično skladište (to je najčešće bio SQL). Ovo je ograničavalo podršku za nove formate podataka i nove načine obrade.



Slika 4. Tradicionalna arhitektura skladišta podataka [1]

1.3.1 Raznovrsnost podataka

Raznovrsnost (*variety*) podatka koji se obrađuju je veoma bitna za analitiku. Tradicionalna skladišta podataka su prvenstveno namenjena za rad sa strukturiranim podacima. Ovo je dobro funkcionisalo dok se većina podataka prikupljala iz relacionih baza podataka, ali sa eksplozijom SaaS ponude, raznih IoT senzora i društvenih mreža, podaci neophodni za analitiku su mnogo raznovrsniji i sve više su zahtevani nestrukturirani oblici podataka kao što su tekst, zvuk, video ili slike.

Sa razvojom SaaS aplikacija i mikroservisnih arhitektura, dolazi i do uspona JSON (*javascript object notation*) formata. Ovaj format iako ima određenu strukturu, ta struktura nije zagarantovana i podložna je promenama. Često se dešava da SaaS dobavljač promeni API (aplikativni programski interfejs) koji nude svojim klijentima i JSON odgovori promene svoju šemu. Tradicionalna skladišta podataka su veoma osetljiva po ovom pitanju i oni su napravljeni tako da podaci moraju da eksplicitno poštuju unapred određenu šemu. Ovaj nedostatak fleksibilnosti i usmerenost samo na strukturirane tipove podataka ograničava organizacije i sprečava razvoj analitičkih sistema.

1.3.2 Količina podataka

S ozbirom na enorman rast količine (*volume*) podataka koja se generiše svakog dana, većina organizacija ima problem da se izbori sa terabajtima i petabajtima novih podataka. U današnjem IT svetu, čak i kompanija srednje veličine mora da bude u stanju da obrađuje na petabajte podataka. Neki od većih izvora novih podataka su: aktivnost korisnika na veb sajtovima, aktivnost na društvenim mrežama, IoT senzori, pametni uređaji i ostali [1].

U tradicionalnim implementacijama skladišta podataka su skladište i obrada čvrsto spregnuti, što znatno ograničava fleksibilnost i skalabilnost [1]. Prilikom povećanja saobraćaja i količine podataka, neophodno je kupiti nove i snažnije servere, sa više radne memorije, procesorske snage i više diskova. S obzirom da ne možemo dobiti više prostora na disku bez da povećamo i procesorsku snagu, to znači da na kraju više plaćamo za nepotrebnu procesorsku i memorijsku snagu. Ova činjenica je značila da je obrada zaista velike količine podataka sa tradicionalnim skladištem podataka, bila namenjena samo za kompanije sa velikim budžetima za informacione tehnologije [1].

1.3.3 Brzina podataka

Brzina (*velocity*) kojom podaci pristižu u platformu možda ne predstavlja veliki problem većini firmi danas, ali ako pogledamo činjenicu da analitika sve više teži da bude u realnom vremenu, samo je pitanje dana kada će postojeće, beč (*batch*) orjentisane arhitekture za analitiku naići na velike probleme. Sa rastućim brojem IoT senzora, strimovanje podataka postaje uobičajeno. Ovaj porast proizvodnje podataka utiče na potrebe za poboljšanjem servisa za prikupljanje strimovanih podataka kao i obradu i analizu tih podataka u što je više moguće realnom vremenu.

Tradicionalna skladišta podataka su više beč orjentisana. Obično su zakazani poslovi koji se izvršavaju po noći i traju nekoliko sati kako bi izračunali analitike za prethodni dan. Strimovani podaci primoravaju organizacije da menjaju postojeće arhitekture platformi tako da budu u stanju da obrađuju svaki novi podatak kako on bude pristizao.

1.4 Revolucija računarstva u oblaku

Računarstvo u oblaku predstavlja isporuku računarskih resursa i skladišnih kapaciteta kao uslugu za grupu krajnjih korisnika. Koncept računarstva u oblaku se oslanja na deljenje resursa preko mreže, najčešće interneta. Krajni korisnici pristupaju aplikacijama u oblaku preko veb pretraživača, desktop aplikacije, mobilne aplikacije, ili programski - kroz upotrebu određenih biblioteka i frejmvorka, dok se softver i korisnički podaci nalaze na serverima na udaljenoj lokaciji. Jedna od glavnih prednosti računarstva u oblaku jeste njegov cenovni model, gde korisnici plaćaju samo onoliko koliko su zaista potrošili koristeći servise u oblaku.

Pre razvoja računarstva u oblaku, kompanije su morale same da obezbede i kupe računarsku opremu (najčešće serveri) koji su im bili potrebni za izvršavanje svog softvera, čuvanje korisničkih podataka ili za druge svhre. Serveri koje bi kompanija morala da kupi, bi se kasnije instalirali i organizovali u sklopu centra podataka (*data center*). Centar podataka zahteva velika početna ulaganja kao i visoku cenu održavanja, počev od struje neophodne za rad servera i njihovo hlađenje, preko osoblja koje održava te servere i na kraju do tehnologije i ljudi zaduženih za fizičku bezbednost centra podataka.

Revolucija koju je donelo računarstvo u oblaku je zaista imala veliki uticaj na celokupnu industriju. Umesto da firme kupuju sopstvenu infrastrukturu i opremaju svoje centre podataka, sada one mogu da iznajmljuju sve vrste servisa koje su im potrebne od strane provajdera. Glavna prednost je to što startapi i manje firme koje tek razvijaju svoj biznis mogu zaobići početna ulaganja i kompleksnost održavanje sopstvene IT infrastrukture i jednostavno plaćati samo za one servise u oblaku koje su zapravo koristili i koliko dugo su ih koristili.

Razvoj javnog računarstva u oblaku doneo je velike prednosti [1]:

- **Elastičnost resursa** - Servisi za obradu (procesiranje), za skladištenje podataka ali i za mnoge druge svrhe, danas mogu da se iznajme od klaud provajdera tako što kompanija izabere tačno određenu količinu snage koja im je potrebna i onda u skladu sa potrebama se taj resurs smanjuje ili povećava automatski na zahtev korisnika.

- **Modularnost** - Skladištenje i obrada podataka su konačno razdvojeni u oblaku. Nema više potrebe da se plaća istovremeno za čuvanje i obradu podataka. Ova činjenica omogućuje optimizaciju troškova prema sopstvenim potrebama.
- **Plaćanje po upotrebi** - Ranije su kompanije koje imaju svoje centre podataka u većini slučajeva plaćale za servere koji su kupili a koji dosta vremena ništa ne radi. Ovo nije slučaj u klaudu, gde kompanije plaćaju samo za ono što su zaista koristili.
- **Samoodrživi servisi postaju nova norma** - U centru podataka su morali da budu zaposleni ljudi koji bi održavali servere i servise, tako što bi instalirali nove verzije softvera, vršili migracije sa pokvarenih mašina i slično. U oblaku su stvari mnogo lakše i postoji mnogo servisa gde provajderi rade većinu infrastrukturnih zadataka umesto nas.
- **Momentalna dosupnost** - Naručivanje i priprema novog servera je nekada ranije znala da traje i po nekoliko meseci. Naručivanje i priprema klaud servisa traje par minuta.
- **Nova generacija alata** - Postoji niz inovativnih klaud servisa koje organizacijama daju pristup tehnologijama koje su ranije bile dostupne samo najvećim igračima.
- **Brzi razvoj novih funkcionalnosti** - Na svakom servisu u oblaku radi veliki broj stručnjaka koji su zaduženi za konstantno unapređenje datog servisa i pobošljavanje korisničkog iskustva. Velika konkurencija na tržištu klaud provajdera predstavlja veliku prednost za krajnje korisnike, jer baš zbog te konkurentnosti, provajderi nastoje da dodaju nove funkcionalnosti i servise kao i da smanjuju cene postojećih servisa.

Jedan konkretan primer na kojem se može videti napredak koji je donelo računarstvo u oblaku jeste *Google BigQuery* [15], moderno skladište podataka u oblaku. Za razliku od tradicionalnih skladišta podatka koji imaju čvrsto spregnute kapacitete za obradu i čuvanje podataka, *BigQuery* kao moderna alternativa u potpunosti menja ovu paradigmu. U ovom servisu su skladište i obrada podataka skroz odvojeni koncepti i tako se mogu koristiti i skalirati po potrebi.

Jedna od glavnih karakteristika ovog servisa koja ga zapravi svrstava u najmodernije klaud servise jeste što korisnici zapravo nisu ni svesni infrastrukture na kojoj se ovaj servis izvršava. Iz ugla korisnika desetine i stotine servera koji su potrebni za izvršavanje jednog *BigQuery* upita nad našim skladištem su zapravo briga klaud provajdera. U ovom slučaju *Google* je na sebe preuzeo obavezu da održava te servere kao i da u svakom trenutku odredi tačan broj servera neophodnih za određenu obradu ili za čuvanje podataka. Ovaj koncept se naziva *serverless* (u prevodu sa engleskog jezika “bez servera”), gde krajnji korisnici imaju tu privilegiju da se isključivo fokusiraju na biznis logiku i direktne doprinose za svoju organizaciju, a ne na održavanje infrastrukture.

Cena za usluge korišćenja ovog servisa jeste po upotrebi, pri čemu se skladištenje i obrada odvojeno naplaćuju. Korsinici plaćaju za onoliko gigabajta podataka koji su trenutno uskladišteni. Kapaciteti za obradu se naplaćuju posebno, a glavni udeo u ceni obrade jeste sama složenost upita, odnosno koliko kapaciteta za obradu je potrebno za njegovo izvršavanje i koja količina podataka je obrađena iz skladišta.

Revolucija računarstva u oblaku verovatno se najviše oseća u domenu skladištenja, obrade i analize velike količine podataka. U tradicionalnim postavkama, svaki susret sa velikom količinom podataka je podrazumevao i velike troškove. Pametnom upotrebom adekvatnih servisa u oblaku, kompanije male i srednje veličine mogu zapravo da otključaju vrednost svojih podataka i da postanu kompanije vođene podacima i činjenicama.

2 Moderna platforma za analitiku

U prethodnom poglavlju smo razmatrali neke od tradicionalnih rešenja za implementaciju sistema za analitiku, počev od jednostavne postavke sa produkcionom bazom u centru sistema do sistema gde je centralna tačka skladište podataka. Uvideli smo da razvoj velikih podataka donosi sa sobom i nove izazove, pa organizacije koje žele da budu lideri u ovom novom dobu moraju da se adaptiraju na rapidni rast podataka i da evoluiraju svoje platforme za analitiku.

Razvoj računarstva u oblaku donosi nove inovacije i alate koji pomažu organizacijama da se izbore sa novonastalim problemima. Jedna od najbitnijih inovacija jeste razvoj modernih skladišta podataka u oblaku kao što su *Google BigQuery*, *Amazon Redshift*, *Azure SQL Data Warehouse* i *Snowflake*. Ova moderna skladišta podataka su napravila svojevrsnu disrupciju na polju analitike i poslovne inteligencije. Skupa, nefleksibilna i neefikasna tradicionalna rešenja su zamenjena novim servisima gde kompanije imaju mogućnost da plaćaju samo za onu količinu podataka koje čuvaju i koje obrađuju, kao i elastičnost koja im omogućava da prilagode performanse sistema svojim poslovnim zahtevima.

U ovom poglavlju ćemo govoriti o tome zašto sistemi za analitiku zasnovani na skladištu podataka ponekad nisu najbolje ili dovoljno dobro rešenje, čak i sa upotrebom modernih tehnologija u oblaku. Predstavićemo alternativno rešenje, platformu za analitiku, kao i njene gradivne blokove i prednosti koje sa sobom nosi ovaj pristup.

2.1 Jezero nasuprot skladišta podataka

Skladište podataka je centralna baza za čuvanje i obradu velike količine podataka za analitičke svrhe. O skladištu se može razmišljati kao o mestu na koje organizacija prebaci sve svoje podatke sakupljene iz raznovrsnih internih sistema. U suštini, skladište podataka nije ništa više nego obična baza podataka optimizovana za potrebe analitike. Moderna skladišta podataka su takozvane MPP (*massive parallel processing*) baze podataka, baze sposobne za masovnu paralelnu obradu podataka na velikom broju mašina. Današnja skladišta u velikoj većini podržavaju SQL interfejs i čuvaju podatke u tabelama sa jasno definisanom šemom i relacionim modelom. Proces učitavanja podataka u skladište predstavlja ekstrakciju podataka iz različitih izvora i učitavanje tih podataka u odgovoarajuće tabele u skladištu. Glavne i osnovne uloge skladišta jesu čuvanje podataka za analitiku i obrada tih podataka.

Osim skladišta podataka, u poslednjih nekoliko godina sve više postaje popularniji koncept jezera podataka (*data lake*). U najosnovnijem obliku, jezero podataka je spremište za čuvanje ogromne količine sirovih i neobrađenih podataka u svom izvornom formatu. Shodno ovim definicijama, u svom početnom periodu razvoja, jezero podataka je predstavljalo mesto gde bi kompanije učitavale sve svoje podatke iz svih svojih sistema u sirovom obliku, bez dodatne obrade. Za razliku od skladišta, gde se čuvaju strukturirani podaci u relacionim tabelama, u jezeru se skladište različiti tipovi podataka: strukturirani, polustrukturirani i nestrukturirani. Jezero možemo posmatrati kao spremište za različite tipove fajlova i objekata, gde podatke možemo čuvati u raznim formatima kao što su JPG, PNG, MP3, MP4, JSON, CSV, TXT, Parquet, Avro i mnogi drugi. Sa ovom fleksibilnošću, jezero podataka ima potencijal da postane

mesto na kojem se mogu grupisati i organizovati celokupni podaci jedne organizacije, bez ograničenja na format tih podataka.

Osim fleksibilne šeme koja omogućava čuvanje raznih tipova podataka, još jedna prednost jezera podataka u odnosu na skladište jeste i cena čuvanja podataka. Moderna jezera podataka u oblaku kao što su *Amazon S3* i *Google Cloud Storage* pružaju veoma povoljne cene i omogućavaju kompanijama da čuvaju petabajte podataka u svojim jezerima podataka sa podnošljivom cenom. U većini slučajeva, iako bi svi podaci jedne organizacije bili strukturirani i pogodni za čuvanje u skladištu podataka, to verovatno ne bi bilo praktično iz finansjiskih razloga, čak i u modernom skladištu u oblaku.

Ranije smo naveli da skladište podataka ima fiksnu šemu po kojoj se podaci čuvaju u jasno definisanim tabelama. Ovo može da predstavlja problem ukoliko imamo izvor podataka čija šema je podložna promenama. Ukoliko kao izvor podataka koristimo određeni API (aplikativni programski interfejs) koji na zahtev vraća podatke u JSON formatu, s obzirom da ovaj format spada u polustrukturirane tipove, može se desiti da se šema odgovora promeni i da se promene tipovi podataka za određena polja. Ukoliko želimo da i dalje učitavamo podatke iz ovog izvora u skladište podataka, moraćemo dodati određene korake transformisanja i obrade izvornih podataka pre skladištenja. Ovaj korak neophodne obrade pre ubacivanja u skladište podataka često onemogućuje kompanije da čuvaju sve svoje sirove podatke u skladištu. Drugim rečima rečeno, skladište podataka ima jasno definisanu šemu koja se mora poštovati i znati prilikom upisivanja novih podataka u njega. Ovaj koncept je poznat kao *schema-on-write*. Nasuprot ovome, jezero podataka nam omogućava da čuvamo podatke u različitim formatima i ne postoji šema koja bi ograničila dodavanje novih podataka, što omogućava lakše dodavanje novih sirovih podataka u skladište. Ovaj način čuvanja podataka nas obavezuje da prilikom obrade tih fajlova koristimo neki servis, alat ili frejmvork za obradu koji će da odredi šemu podataka prilikom čitanja fajla iz jezera. Ovaj pristup je poznatiji kao *schema-on-read*.

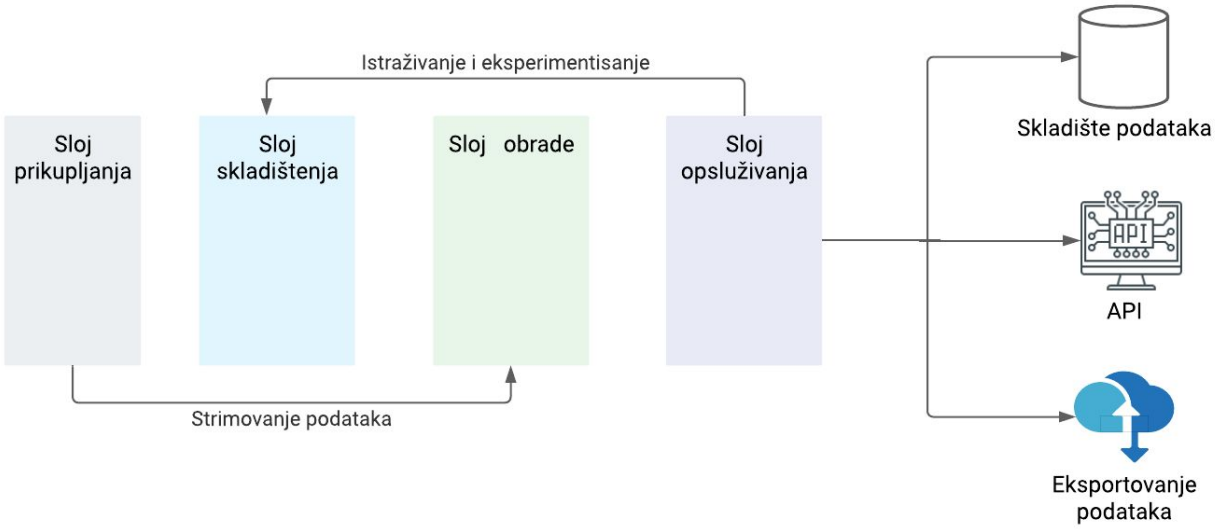
U skladištu podataka najčešće se nalaze obrađeni podaci spremni za upotrebu. Da bi omogućili korišćenje i istraživanje ovih podataka širem krugu ljudi unutar organizacije, obično se koristi alat za vizuelizaciju i poslovnu inteligenciju koji se direktno integriše sa skladištem podataka. Ovi alati omogućavaju netehničkom osoblju kao što su biznis analitičari i rukovodici kompanije da imaju pristup i uvid u podatke iz skladišta podataka kroz razne vizuelizacije, grafikone i slično. Ovo znači da je uloga skladišta podataka da omogući svakom pojedincu u organizaciji da donosi odluke zasnovane na činjenicama i informacijama koje se nalaze u skladištu. Sa druge strane, podaci u okviru jezera podataka obično nisu obrađeni u tolikoj meri i često su u sirovom izvornom obliku. Najčešći korisnici tih podataka iz jezera su tehničko osoblje, poput inženjera obrade ili analize podataka koji pristupaju tim podacima upotrebom nekog programskog jezika ili alata za obradu podataka kako bi ih transformisali u pogodniji oblik ili direktno koristili za izvršavanje određenog zadatka. U tabeli ćemo sumirati razlike između jezera i skladišta.

	Skladište podataka	Jezero podataka
Podaci	Stuktuirani, obrađeni	Strukturirani, polustrukturirani, nestrukturirani, neobrađeni
Format	Relacioni model, tabele	TXT, CSV, JSON, JPG, *Parquet*, i drugi
Obrada	*Schema-on-write*	*Schema-on-read*
Skladište	Skupo za čuvanje velike količine podataka	Dizajnirano za jeftino čuvanje podataka
Agilnost	Manje agilno, fiksirana konfiguracija	Veoma agilno, konfiguracija promenljiva
Korisnici	Biznis analitičari, rukovodioci	Inženjeri obrade i analize podataka

Tabela 1. Poređenje skladišta i jezera podataka [14]

2.2 Gradivni blokovi platforme

Glavna svrha platforme za analitiku jeste da prikuplja, skladišti, obrađuje podatke i čini ih dostupnim za dalju analizu bez obzira na tip podataka koji dolaze i to na najisplativiji mogući način. Da bi platforma bila u mogućnosti da ispuni ove uslove, njena arhitektura ume često da bude dosta složena. Optimalna arhitektura platforme za analitiku se postiže podelom na slojeve arhitekture, gde je svaki sloj zadužen za izvršavanje tačno jedne specifične funkcije. Osnovni gradivni blokovi platforme za analitiku su sloj prikupljanja, sloj skladištenja, sloj obrade i sloj opsluživanja. U nastavku ćemo za svaki od slojeva detaljnije opisati njegove odgovornosti.



Slika 5. Arhitektura platforme za analitiku [1]

2.2.1 Sloj prikupljanja

Osnovna odgovornost sloja za prikupljanje podataka jeste sakupljanje podataka u platformu. Njegova odgovornost je da ekstraktuje podake iz raznovrsnih izvora kao što su relacione baze ili NoSQL baze podataka, fajl skladišta, interni ili eksterni aplikativni programski interfejs. Sa

razvojom tehnologije, došlo je i do velikog porasta broja različitih izvora podataka iz kojih organizacije trebaju da sakupe podatke za svoju analitiku, i stoga ovaj sloj mora da bude veoma fleksibilan i nadogradiv. Progarmeri se često odlučuju da za implementaciju ovog sloja upotrebe neke od dosupnih *open-source* ili komercijalnih alata.

Jedna od najbitnijih karakteristika sloja za prikupljanje jeste da on ne sme ni u kom slučaju da modifikuje ili transformiše nadolazeće podatke. Dobra praksa je da se podaci dovuku u jezero podataka u izvornom, neobrađenom obliku [1].

2.2.2 Sloj skladištenja

Nakon što smo prikupili podatke iz različitih izvora, moramo na neki način da ih sačuvamo. Kao što je navedeno u prethodnim poglavljima, jezero podataka je dobar izbor za skladištenje neobrađenih podataka. Skalabilnost i ekonomičnost jezera podataka nam omogućava da sačuvamo sve sirove podatke koje dolaze u velikim količinama i velikom brzinom, kao i sve nove podatke koji su generisani izvođenjem transformacija nad sirovim podacima.

Moderna jezera podataka se danas obično implementiraju upotrebom nekih od servisa u oblaku namenjenih za ovaj scenario. Popularna rešenja koja se danas koriste za ove potrebe su *Amazon S3* i *Google Cloud Storage*. Ova moderna jezera podataka nemaju nikakva ograničenja što se tiče tipova fajlova koji treba da se sačuvaju. Mogućnost da se sačuva bilo koji fajl format predstavlja osnovu za jezero podataka, jer na ovaj način možemo prvo da sačuvamo neobrađene podatke u izvornom formatu i da obradu ostavimo za kasnije. Svi veliki klaud provajderi danas imaju u svojoj ponudi servis za skladištenje koji može da se koristi kao jezero podataka. Glavne prednosti klaud skladišta su [1]:

- Provajderi vode računa o infrastrukturi, osvežavanju softvera, skaliranju servera i održavanju servisa, a krajnji korisnici mogu da se fokusiraju samo na čuvanje i upotrebu svojih podataka u okviru skladišta.
- Klaud skladište je elastično. Ovo znači da ne moramo unapred da izaberemo velilčinu skladišta u nekoj jedinici mere, već su provajderi zaduženi da skaliraju servis za bilo koju veličinu podataka koja nam je u trenutku potrebna.
- Korisnici plaćaju samo za onu količinu podatka koja je trenutno uskladištena.
- Klaud skladište omogućuje razdvajanje skladištenih kapaciteta od kapaciteta za obradu.

2.2.3 Sloj obrade

Nakon što smo učitali podatke u platformu i sačuvali ih u svom izvornom formatu, vreme je da obradimo i transformišemo podatke u neki drugi oblik od kojeg ćemo imati više koristi. Implementacija sloja za obradu podataka je obično najinteresantniji korak pri dizajniranju jezera podataka i platforme za analitiku. S obzirom da su podaci već sačuvani u jezeru podataka, moguće je izvršavati direktnu analizu nad tim neobrađenim podacima, ali u većini slučajeva to nije najefektivniji i najproduktivniji način. Obično se organizacije odlučuju da te podatke transformišu u oblik pogodniji za analizu i upotrebu od strane analitičara i inženjera.

Za razliku od skladišta podataka, gde smo u većini slučajeva ograničeni na pisanje transformacija u SQL jeziku, za obradu podataka u jezeru podataka postoji veliki broj programskih jezika, alata i frejmvorka koji se mogu upotrebiti. Moguće je takođe korsititi SQL transformacije u okviru jezera podataka i to uz pomoć alata kao što su *Hive* i *Presto*, koji u

suštini predstavljaju viši nivo apstrakcije jezera podataka u obliku skladišta podataka i omogućavaju lak način za manipulaciju i obradu fajlova. Prilikom upotrebe ovih alata, korisnik ima utisak da radi u okruženju sličnom standardnog skladišta podataka gde su podaci sačuvani u tabelama, dok su zapravo svi podaci sačuvani u fajlovima u nekom od formata.

Iako je SQL veoma rasprostarnjen jezik za pisanje transformacija nad podacima, on nije naročito robustan programski jezik. Teško je ekstraktovati zajedničku logiku i ponavljajuće korake čišćenja podataka u posebno ponovo upotrebljivu biblioteku u okviru SQL-a. Modularnost i apstrakcija koja krasi programske jezike nedostaje SQL-u. U poslednjih nekoliko godina razvili su se mnogobrojni frejmvorci za obradu podataka koji kombinuju skalabilnost i paralelizam sa potencijalom modernih programskih jezika. Neki od najznačajnijih ovakvih alata su *Apache Spark*, *Apache Beam* i *Apache Flink* [1].

2.2.3 Sloj opsluživanja

Cilj sloja za opsluživanje podataka jeste da pripremi podatke za konzumiranje od strane krajnjih korisnika, bilo da su to ljudi ili drugi sistemi. Sve veći broj ljudi, sa različitim nivoom iskustva i znanja, koji zahtevaju pristup podacima u okviru organizacije predstavljaju veliki izazov za platformu za analitiku. Pored različitog nivoa tehnološkog iskustva, ovi ljudi najčešće imaju i različite preference što se tiče alata koje upotrebljavaju.

Biznis korisnici, menadžeri i rukovodioci kompanija žele pristup izveštajima, grafikonima i vizuelizacijama podataka uz veliku mogućnost samousluživanja prilikom navigacije ovim alatima, kako ne bi ovi korisnici morali stalno da zahtevaju nove podatke od inženjera. Njima je obično potreban alat za poslovnu inteligenciju koji će im omogućiti da nezavisno upoređuju i istražuju sačuvane podatke. Biznis korisnici obično zahtevaju da se podaci prikazuju veoma brzo, što znači da upiti koji se izvršavaju nad našim podacima moraju brzo da se završe. Ovo je jedan od razloga zbog čega se određeni deo podataka iz jezera transformiše i učitava u skladište za brzo izveštavanje.

Inženjeri obrade i analize podataka s obzirom na svoje tehnološtko iskustvo i znanje i skladno svojim preferencama što se tiče alata, mogu da izaberu koji će alat da koriste za obradu i analizu podataka. Oni mogu da koriste skladište koje ima obrađene i transformisane podatke, ali takođe mogu da se opredele za opciju da koriste podatke direktno iz jezera, jer možda žele da isprobaju neki novi način upotrebe neobrađenih podataka.

2.3 Platforma nasuprot 3V

Ranije smo razmotrili neke od problema sa tradicionalnim arhitekturama sistema za analitiku koji se pojavljuju kada organizacije pređu se male ili srednje količine podataka na velike podatke. U nastavku razmatramo kako se platforma za analitiku nosi sa ovim problemima i kako ih rešava.

2.3.1 Raznovrsnost

Platforma za analitiku u oblaku je u stanju da prihvati različite tipove podataka zbog svoje slojevite arhitekture. Sloj za prikupljanje podataka može biti implementiran kao kolekcija alata, gde se svaki od njih brine za tačno jedan izvor podataka. Druga opcija za implementaciju sloja

prikupljanja jeste da se jedna aplikacija bavi sakupljanjem, ali da bude ekstenzibilna tako da se mogu dodavati ili uklanjati nastavci za različite izvore podataka. Jezero podataka kao što je klaud skladište je u stanju da prihvati bilo koji format podataka zato što je ono zapravo generički fajl sistem, što znači da možemo da sačuvamo podatke kao što su JSON, CSV, video, audio i ostali tipovi. Ne postoji nikakva ograničenja što se tiče tipova podataka koji se čuvaju u jezeru.

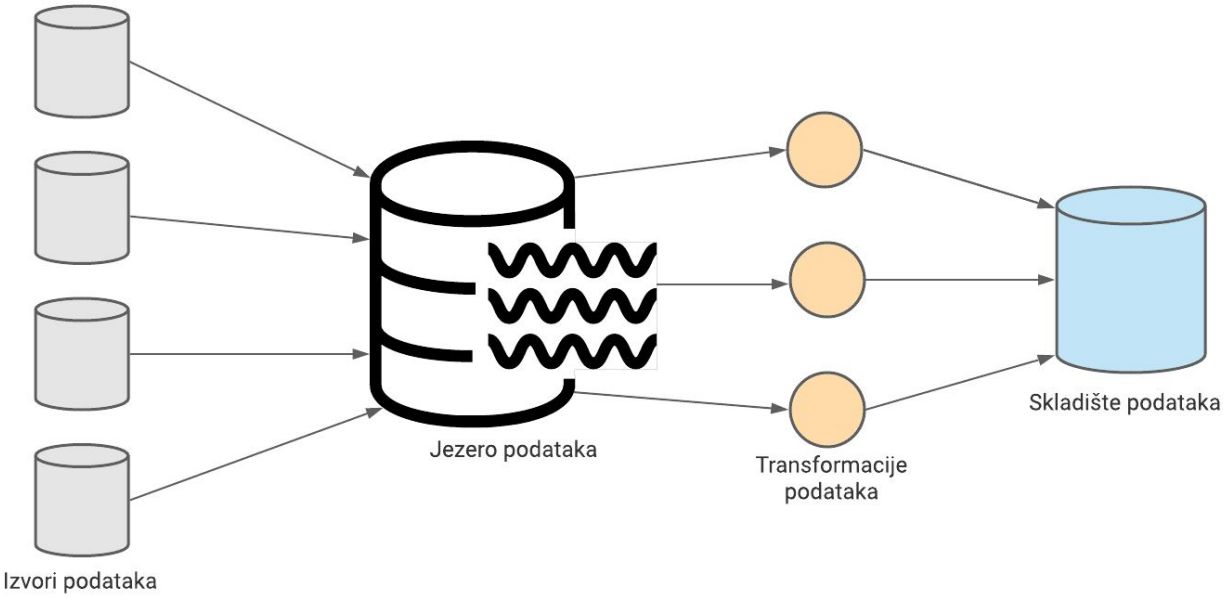
Postojanje novih alata za obradu podataka kao što su *Apache Spark* i *Apache Beam*, oslobađaju nas ograničenja sa kojima su se organizacije suočavale kada bi im bila dostupna samo SQL obrada podataka.

2.3.2 Količina

Klaud provajderi danas pružaju usluge prikupljanja, čuvanja, obrade i analize velike količine podataka bez troškova unapred, već organizacije plaćaju samo za one resurse koje su zapravo koristili ili potrošili. Odvajanje sloja za obradu i sloja za skladištenje kao i fleksibilan cenovni model u kojem korisnici plaćaju samo ono što potroše su omogućili da obrada podataka u oblaku bude jednostavnija i jeftinija nego u tradicionalnim sistemima.

U domenu servisa za obradu podataka, pojavili su se novi i inovativni klaud servisi koji korisnicima pružaju priliku da obrade nekada nezamislive količine podataka. Elastičnost i dostupnost servisa za obradu omogućavaju korisnicima da se resursi kreiraju samo kada su neophodni, kao i da se obrišu kada su završili zadati posao. Na ovaj način kompanije ne plaćaju za kapacitete obrade koji zapravo ništa ne rade.

Iako je skladištenje podataka u jezeru podataka skoro uvek najjeftinija opcija, današnji de fakto standard za obrađene podatke koji su neophodni biznis korisnicima i analitičarima jeste da se oni nalaze u skladištu podataka koje omogućava ekstremno brze performanse pri analizi [1]. Deo sirovih i neobrađenih podataka iz jezera se kroz seriju transformacija učitava u skladište podataka gde postaje dostupan za efikasnu poslovnu analizu i prikaz.



Slika 6. Arhitektura platforme sa jezerom i skladištem podataka

2.3.1 Brzina

Moderne aplikacije u današnjem svetu sve više imaju prediktivne funkcionalnosti koje se izvršavaju u realnom vremenu, poput predviđanja najbolje ponude za aktivnog korisnika.
Moderna platforma za analitiku treba da dozvoli integraciju i koegzistenciju prikupljanja i analize strimovanih podataka u realnom vremenu i beč orjentisanog izveštavanja i tradicionalne poslovne inteligencije. Klaud provajderi u svojoj ponudi imaju servise za obradu podataka u realnom vremenu koji mogu da zaobiđu relativno sporo klaud skladište i da šalju podatke direktno u sloj za obradu i analizu da bi postigli zahtevane performanse.

Sa kapacitetima za obradu koji su dostupni na zahtev u oblaku, ne postoji više potreba da sistemi koji rade u realnom vremenu i beč sistemi dele resurse. Sada mogu da postoje odvojeni resursi za obe vrste obrade i analize podataka. Nakon što dobije nove podatke, sloj za obradu može da ih pošalje na različite destinacije kao što su brza baza podataka koju koristi aplikacija u realnom vremenu, jezero podataka za arhiviranje ili skladište podataka za izveštavanja i poslovnu inteligenciju.

2.4 Prednosti platforme za analitiku

Razumevanje potencijalnih prednosti i slučajeva korišćenja platforme za analitiku podataka je veoma bitno prilikom dizajniranja nove platforme. Bez ovog konteksta, postoji mogućnost da organizacija napravi kompleksnu platformu koja zapravo ne donosi prave poslovne vrednosti.

Jedan od glavnih koristi koje organizacije žele da dobiju od platforme za analitiku jeste mogućnost da imaju pogled od 360 stepeni na svoje korisnike i poslovanje firme. Korisnici komuniciraju sa kompanijom preko različitih medijuma kao što su mobilne aplikacije, veb aplikacije, društvene mreže, kanali za korisničku podršku i drugi. Podaci iz ovih interakcija mogu da budu i strukturirani i nestrukturirani, mogu da budu različitog kvaliteta, veličine i da se generišu različitom brzinom. Integracija i konsolidacija svih tački interakcije korisnika sa kompanijskim sistemima otvara vrata novim poslovnim prilikama kao što su poboljšano korisničko iskustvo, bolja personalizacija u marketingu, manje korisnika koji napuštaju kompanijske sisteme i unapređenje poslovanja u svakom aspektu.

Drugi slučaj upotrebe jezera podataka jeste za skladištenje IoT (internet od stvari) podataka. Na ovaj način, podaci sa senzora raznih pametnih uređaja i mašina se konsoliduju na jedno mesto i njihovom obradom može se doći do uvida u rad i stanje tih istih uređaja. Podaci koji dolaze sa senzora su obično veoma veliki količinski, dolaze u veoma kratkim intervalima ali su veoma nesigurni, jer senzori ponekad nemaju pristup internetu, pa se dešavaju kašnjenja. Ove osobine čine jezero podataka dobrim mestom za čuvanje IoT podataka.

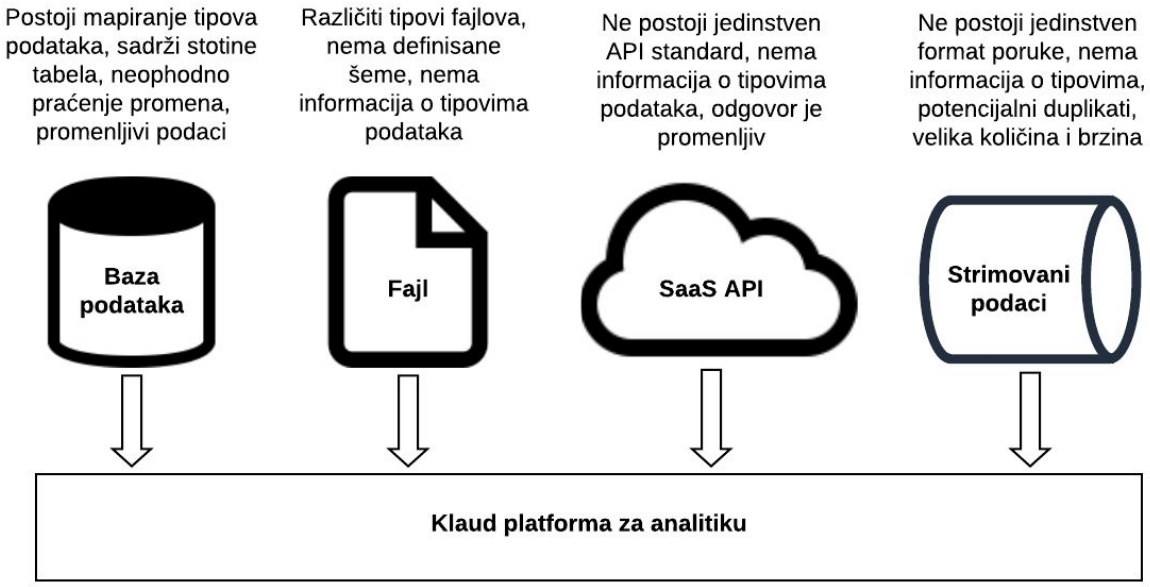
Razvoj napredne analitike upotrebom mašinskog učenja i veštačke inteligencije je povećao adopciju jezera podataka u većini arhitektura platforme za analitiku iz razloga što te tehnike zahtevaju ogromne količine podataka, obično mnogo veće nego što bi bilo ekonomski isplativo ako bi te podatke čuvali i obrađivali u sklopu skladišta podataka. Sposobnost jezera podataka da sačuva skoro neograničene količine sirovih podataka na ekonomičan način i sposobnost platforme da obrađuje ove podatke bez uticaja na performanse krajnjih aplikacija predstavlja veliku prednost.

3 Prikupljanje i centralizovanje podataka

U ovom poglavlju ćemo detaljnije istražiti slojeve prikupljanja i čuvanja podataka. Prvi korak koji je neophodan u implementaciji klaud platforme za analitiku jeste implementacija sloja za prikupljanje kako bi podaci počeli da pristižu u platformu. Jedna od ključnih karakteristika platforme jeste sposobnost da se izbori za heterogenim izvorima koji generišu podatke u različitim oblicima i formatima [1]. Raznovrsnost izvora podataka potrebno je dobro razmotriti pre same implementacije i u ovom poglavlju ćemo se baviti specifičnostima prikupljanja podataka iz nekoliko različitih izvora kao što su baze podataka, fajlovi, aplikativni programski interfejsi (API) i strimovi. Nakon što se napravi adekvatna podrška za prikupljanje podataka iz različitih izvora, neophodno je te podatke centralizovati i uskladištiti u zajedničko spremište.

3.1 Izvori podataka

Svaki izvor podataka ima svoje specifične karakteristike. Na primer, relacione baze podataka uvek imaju tip povezan sa određenom kolonom u tabeli, dok recimo u CSV (*comma separated value*) fajlovima koji imaju oblik tabele, nemamo asocirane tipove podataka, već je zadatak aplikacije koja čita fajl da to odredi. Interfejsi koje nude SaaS aplikacije su najčešće podešeni da vraćaju odgovor u JSON formatu. Ovaj format je polustrukturiran pa se mogu očekivati određene izmene koje treba adekvatno obraditi. U narednom dijagramu su ilustrovane glavne karakteristike ovih izvora podataka. Strimovani podaci predstavljaju posebnu kategoriju koja je doživela ekspanziju u protekloj deceniji [1]. To su podaci koji se generišu velikom brzinom i primoravaju organizacije da razmišljaju o adekvatnim načinima za prihvatanje tih brzo nadolazećih podataka kao i njihovu obradu u realnom vremenu.



Slika 7. Raznovrsnost izvora podataka [1]

3.1.1 Baza podataka kao izvor

Organizacije koje većinu svog poslovanja obavljaju preko veb sajta ili aplikacije (poput *Amazona* na primer), imaju veliku količinu operativnih i transakcionih podataka. Ove aplikacije su direktno povezane sa produkcionom bazom podataka, koja predstavlja centralno mesto gde se čuvaju svi transakcioni podaci neophodni za pravilan rad aplikacije i ostalih internih sistema. Aplikaciona baza podataka može biti SQL baza ili NoSQL baza podataka.

Relacione baze podataka su jedan od najčešćih izvora iz koji se prikupljaju podataci za analitičku platformu. Podaci u ovim bazama su organizovani u tabele, gde svaka tabela sadrži jednu ili više kolona, a svaka kolona ima svoj tip podataka. Ova jasno i precizno definisana šema podataka obezbeđuje da se ne desi slučaj ubacivanja podatka u pogrešnu tabelu ili kolonu. Tabele u okviru baze su često normalizovane, što znači da su logički entiteti razbijeni na više fizičkih tabela koje su međusobno povezani primarnim i stranim ključevima [1]. Zbog ovog načina modelovanja tabela, nije toliko neobično da jedna relaciona baza sadrži stotine tabela [1]. S obzirom da se relacione baze koriste kao baze za podršku poslovanju, one se često menjaju, svaki događaj u okviru aplikacije mora da se preslika u neke izmene u bazi.

Kada razmatramo dovlačenje podataka iz relacionih baza u našu platformu, gotovo sve baze imaju neke zajedničke karakteristike koje valja razmotriti [1]:

- **Mapiranje tipova podataka** - Kolone sa svojim tipovima podataka u relacionoj bazi se moraju mapirati na kolone u okviru skladišta podataka. Nažalost, svaki proizvođač relacionih baza ima svoje specifične tipove podataka. Iako postoje tipovi koji su veoma slični ili isti u svakom sistemu kao što su *string* ili *integer*, takođe postoje specifični tipovi poput *timestamp* ili *date* koji mogu da imaju drugačije značenje, preciznost ili format.
- **Automatizacija** - S obzirom da relacione baze mogu imati i na stotine tabela, potrebno je da proces dovlačenja podataka bude automatizovan i podesiv. Malo je verovatno da će jedna osoba imati dovoljno vremena za manuelno konfigurisanje dovlačenja podataka iz nekoliko stotina tabela. Ako nekim slučajem postoji dovoljno vremena za ručnu postavku, postojaće velika šansa za ljudsku grešku u tom procesu.
- **Promenljivost** - Podaci u bazama su promenljive prirode. Poslovanje jedne kompanije nikad ne miruje i konstantno se dešavaju nove transakcije. Ukoliko posmatramo primer onlajn prodavnice, uvidećemo da se stotine ili hiljade novih porudžbina zakazuje, šalje ili otkazuje svake minute. Ove akcije rezultuju izmenama velikog broja tabela.

3.1.2 Fajlovi kao izvor

Fajlovi su još jedan veoma čest izvor podataka platforme za analitiku [1]. Obično se radi o tekstualnim ili binarnim fajlovima koji se direktno prebacuju na destinaciju preko FTP (*file transfer protocol*) ili se učitavaju u klaud jezero podataka kao što su *Amazon S3, Google Cloud Storage ili Azure Blob Storage* [1]. Fajlovi mogu da se generišu i automatski: serverska aplikacija generiše *log* fajlove u kojima čuva sve bitne događaje za taj dan. Druga opcija jeste ručno generisanje CSV, TXT ili XLS fajlova od strane neke osobe unutar ili van organizacije.

Fajlovi se na prvi pogled čine kao veoma jednostavan izvor podataka, ali sa stanovišta automatizovanja dovlačenja podataka iz fajlova, ne treba ih olako shvatiti i potrebno je detaljno razmotriti njihove karakteristike [1]. Jedna od bitnijih karakteristika jeste to što dolaze u

različitim i mnogobrojnim formati, bilo tekstualnim ili binarnim. Najpopularniji formati koje možemo sresti su CSV, JSON i XML. Binarne formate kao što su *Parquet* i *Avro* srećemo ređe ali oni igraju veoma bitnu ulogu u svetu velikih podataka [1]. Tekstualni fajlovi ne uključuju tipove podataka za kolone i obično ne određuju neku posebnu strukturu unutar samog fajla. Ovo znači da fajlovi koje čitamo sa istog izvora ne moraju da prate istu strukturu, kao i da se mogu menjati kroz vreme. Sloj za prikupljanje podataka mora biti spreman da odgovori na moguće promene u izvornim fajlovima kao i dovoljno robustan da se izbori sa svim slučajevima.

Detalji koje je potrebno razmotriti prilikom implementacije sistema za dovlačenje fajlova [1]:

- **Parsiranje različitih formata** - Neophodno je da sistem bude u stanju da parsira različite fajlove poput CSV, JSON, XML, *Avro* i drugi. Za tekstualne fajlove ne mora postojati garancija da će proizvođač koristiti istu strukturu kakvu parser očekuje.
- **Promena šeme** - Za razliku od relacionih baza, dodavanje nove kolone ili promena tipa jedne kolone su jednostavne operacije za onoga ko generiše JSON ili CSV fajlove, pa shodno tome, sistem treba da bude u stanju da reaguje pravilno na promenu šeme fajla.
- **Snimak i višestruki fajlovi** - Za razliku od relacionih baza koje su promenljive, fajlovi predstavljaju snimak stanja podataka u vremenu. Ustaljen redosled događaja je sledeći: podaci se ekstraktuju iz izvornog sistema, zatim sačuvaju u fajl i na kraju učitaju u platformu. Ovi snimci izvornog sistema mogu biti u obliku jednog ili više fajlova.

3.1.3 Strim kao izvor

Jedan od novijih izvora podataka koji su potrebni platformama za analitiku jeste strim podataka. Strim podataka predstavlja događaje koji se generišu u velikim količinama i pri velikom brzinom. Koncept strimovanja podataka postaje sve više zastupljen u današnjem svetu i sve veći broj organizacija ima potrebu da analizira događaje i reaguje na njih u realnom vremenu [1]. Primer za strimovane podatke može biti mobilna igrica, gde se svaka akcija i klik korisnika pretvaraju u događaje i šalju ka sistemima za prihvatanje događaja. Drugi primer jesu IoT senzori koji svake sekunde proizvode nove podatke i ta merenja šalju preko mreže na dalju obradu i analizu.

Apache Kafka je verovatno napoznatija platforma za prihvatanje i kratkoročno čuvanje striming podataka [23]. U poslednjih nekoliko godina su se razvili klaud servisi koji se koriste za istu i sličnu namenu kao što su *Amazon Kinesis*, *Google Cloud Pub/Sub*, *Azure EventsHub*. Pri radu sa striming podacima, osim platforme kao što je *Kafka* koja prihvata poruke, postoje proizvođači (*producers*) koji prave nove poruke i šalju ih u strim, i postoje potrošači (*consumers*) koji čitaju poruke iz *Kafke*. Ove tehnologije su organizacijama dale mogućnost da se izbore sa velikim količinama striming podataka. Bitne karakteristike striming podataka su [1]:

- Poruke u okviru striming platforme imaju restrikcije što se tiče formata. Obično se poruke čuvaju kao niz bajtova i mogu nastati kodiranjem JSON, *Avro* ili sličnih formata. Sloj za prikupljanje podataka mora biti u stanju da dekodira te poruke.
- Striming platforme mogu da sadrže i duplicirane poruke što implicira da potrošači treba da budu u stanju da se izbore sa duplikatima.
- Poruke koje se upišu u strim su nepromenljive. Kada se poruka jednom upiše u sistem poput *Kafke*, ona se više ne može izmeniti, ali se može upisati nova verzija te poruke.
- Striming podaci obično dolaze u velikim količinama. Nije neobično da organizacije sakupe milijarde događaja dnevno. Sloj prikupljanja treba da bude skalabilan.

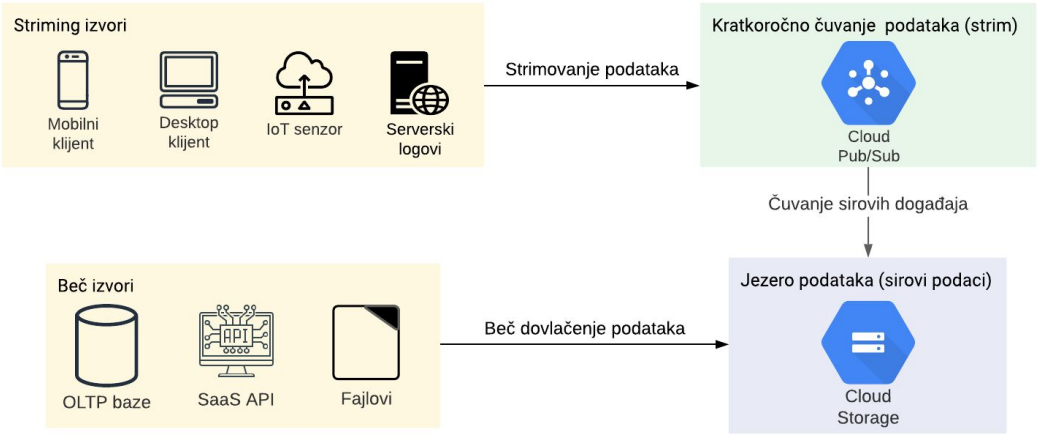
3.1.4 SaaS API kao izvor

Većina organizacija danas koristi bar jedan SaaS (*Software as a Service*) proizvod kao podršku svom poslovanju [1]. Proizvodi poput *Salesforce* i *Marketo* čuvaju neke od najbitnijih skupova podataka za jednu organizaciju [1]. Kombinovanje korisničkih podataka (npr. *Salesforce* podaci) i podataka za marketinške kampanje (npr. *Marketo*) sa transakcionim podacima koje organizacija čuva u svojim produkcionim bazama je jedna od veoma poželjnih prednosti platforme za analitiku. Većina SaaS kompanija omogućava ekstrakciju podataka preko REST (*Representational State Transfer*) API. Obično postoji mogućnost da se podaci ručno preuzmu iz aplikacije u CSV formatu, ali ekstrakcija podataka preko API-ja je u većini slučajeva praktičnija i fleksibilnija. Postoje razni problemi koje srećemo pri korišćenju SaaS API. Neki od najbitnijih su [1]:

- Svaki provajder ima jedinstven način za izlaganje svojih podataka eksternim korisnicima, što znači da ne postoji standard koje sve kompanije implementiraju.
- Većina SaaS API-ja nema informacije o tipovima podataka i šema je podložna promeni.
- Ne postoji jedinstven standard za inkrementalno i potpuno dovlačenje podataka iz SaaS API-ja. Organizacije moraju da se prilagode pojedinačnim provajderima.

3.2 Prikupljanje podataka

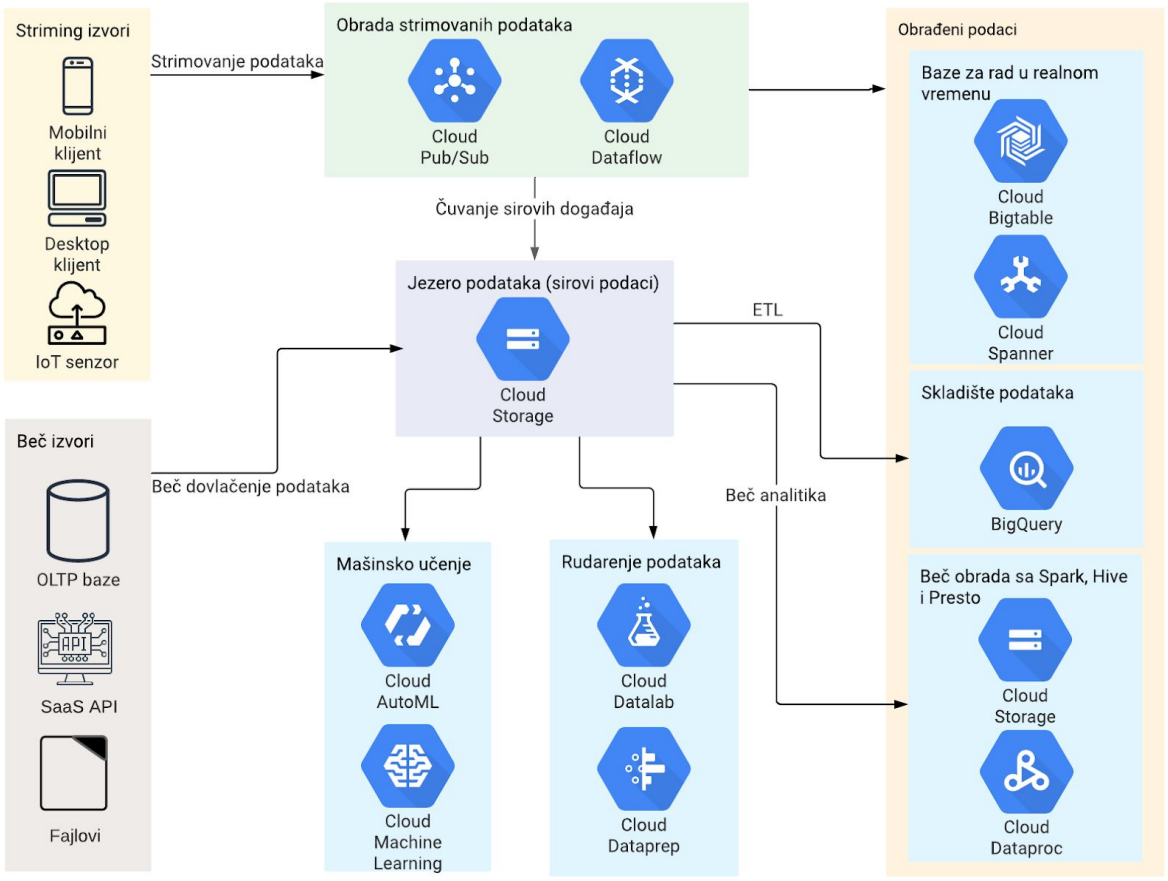
Za implementaciju jezera podataka može se koristiti *Cloud Storage* servis iz *Google Cloud* ponude. Skalabilnost, održivost, ekonomičnost i fleksibilnost *Cloud Storage* servisa čini ga idealnim kandidatom za implementaciju jezera podataka [17]. *Cloud Pub/Sub* je skalabilan i bezbedan servis za razmenu poruka između aplikacija i servisa, potpuno kontrolisan od strane klaud provajdera, koji ćemo mi u našoj platformi koristiti kao servis za prihvatnje i kratkoročno čuvanje striming podataka [25]. *Pub/Sub* prihvata događaje iz izvora kao što su mobilni i veb klijenti, IoT senzori ili aplikativni serveri i ti podaci se čuvaju na kraći vremenski period (obično manje od 7 dana). Potrošači mogu da se zakače za strim, da prihvataju i obrađuju događaje u realnom vremenu. Dobra praksa je da se svi događaji iz *Pub/Sub* proslede u *Cloud Storage* gde se dugoročno čuvaju u neizmenjenom obliku [17]. Na ovaj način postižemo to da se događaji za koje je bitna brza obrada i analiza nalaze u Pub/Sub servisu gde su kašnjenja minimalna, a u jezeru podataka će se čuvati svi istorijski događaji koji mogu poslužiti kao bekap ili kao izvor budućih analiza.



Slika 8. Prikupljanje podataka za platformu

4 Obrada i analiza podataka

Nakon što su uspostavljeni sistemi za prikupljanje i centralizovanje podataka, potrebno je implementirati sloj obrade i analize koji će organizacijama pomoći da od neobrađenih podataka kroz seriju različitih i potencijalno međuzavisnih koraka obrade dođu do krajnjeg cilja, tj. do transformisanih i uređenih podataka, spremnih za razne analize. S obzirom na količinu i raznovrsnost sirovih podataka koji se nalaze u jezeru, potrebno je iskoristiti nekoliko različitih klaud servisa za obradu podataka, gde je svaki od njih namenjen za rešavanje određenog problema. Sloj obrade i analize podataka će u ovom poglavlju biti obrađen kroz detaljnije razmatranje glavnih tokova rada koji su potrebni jednoj organizaciji kao što su [17]: rudarenje i istraživanje podataka, skladištenje podataka i poslovna inteligencija, beč analitika, analitika u realnom vremenu i primena veštačke inteligencije i algoritama mašinskog učenja.



Slika 9. Arhitektura visokog nivoa platforme za analitiku

4.1 Orkestracija poslova

Obrada i analiza u svetu velikih podataka znaju često da postanu veoma kompleksni. Bitna karakteristika svake platforme za analitiku jeste da podaci teku kroz nju, od servisa do servisa, menjajući svoju strukturu i oblik kako bi postali pogodniji za izvođenje završnih analiza. Protok podataka kroz sistem mora da se prati, kontroliše i eventualno ispravi ako dođe do greške.

Da bi se od sirovih podataka preuzetih iz heterogenih izvora došlo do obrađenih i uređenih skupova spremnih za analizu, neophodno je da se izvrši nekoliko različitih i međusobno

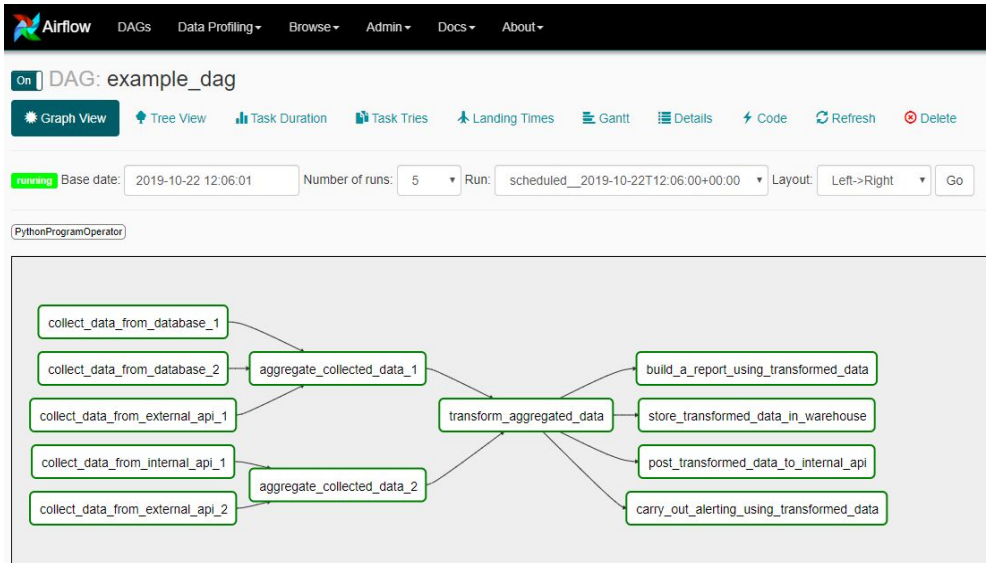
zavisnih koraka. Posmatrajmo primer u kojem organizacija želi da sjedini korisničke podatke (*Salesforce*), marketinške podatke (*Marketo*) i interne transakcione podatke (produkcione OLTP baze) kako bi došla do novih zaključaka. Koraci za izvršavanje ovog procesa su:

1. Dovlačenje *Salesforce* podataka u klaud skladište
2. Dovlačenje *Marketo* podataka u klaud skladište
3. Dovlačenje transakcionih podataka u klaud skladište
4. Čišćenje i agregacija *Salesforce* podataka
5. Čišćenje i agregacija *Marketo* podataka
6. Čišćenje i agregacija transakcionih podataka
7. Sjedinjavanje sva 3 skupa i učitavanje u skladište podataka
8. Izvršavanje validacija za proveru kvaliteta podataka

Kompleksnost ovog procesa proizilazi iz međuzavisnosti različitih koraka. Određeni koraci su nezavisni i mogu se paralelno izvršavati, dok drugi koraci moraju čekati završetak roditeljskih koraka kako bi počeli svoje izvršavanje. Tokovi rada u svetu velikih podataka znaju da budu i znatno složeniji od ovog primera, što je navelo ljude iz industrije na kreiranje softverskog rešenje koje se bavi orkestracijom poslova za dovlačenje, obradu i analizu [1].

Jedan od najpopularnijih alata otvorenog koda za orkestraciju poslova jeste *Apache Airflow*. Nastao je 2014. godine u kompaniji *AirBnb*, gde je korišćen za interne potrebe upravljanja kompleksnim procesima, a kasnije je dobio licencu otvorenog koda [31]. *Airflow* omogućava programsko kreiranje i zakazivanje poslova, kao i nadgledanje i upravljanje kroz korisnički interfejs [31]. Ovaj softver je napisan u programskom jeziku *Python* i kreiranje tokova rada se takođe vrši programski kroz *Python* skripte. Izgrađen je na osnovu principa konfiguracije kroz kod, što je omogućilo programerima da koriste postojeće biblioteke u kreiranju tokova rada [31].

Apache Airflow koristi usmerene aciklične grafove (*Directed Acyclic Graph*, DAG) za upravljanje međuzavisnim poslovima i procesima. Zavisnosti između poslova se definišu u *Python* skripti, a *Airflow* je zatim odgovoran za zakazivanje i izvršavanje poslova [31]. Jedan usmereni aciklični graf je ekvivalent toka rada koji sadrži veći broj međuzavisnih čvorova (poslova). Grafovi se mogu izvršavati, ili na osnovu unapred zakazanog rasporeda (svaki sat ili svaki dan), ili na osnovu nekih eksternih događaja (izmena ili kreiranje novog fajla u klaud skladištu).



Slika 10. Izgled *Apache Airflow* interfejsa i jednog grafa izvršenja

Google Cloud platforma u svojoj ponudi poseduje servis *Cloud Composer* koji je zasnovan na *Apache Airflow*, gde je *Google* preuzeo na sebe odgovornost kreiranja i upravljanja infrastrukturom, kao i instaliranja i održavanja *Airflow* softvera. Krajnji korisnici se mogu fokusirati na kreiranje grafova (tokova rada ili pajplajna). Primeri *Cloud Composer* posla mogu biti: prebacivanje fajla sa udaljenog mesta u *Cloud Storage*, izvršavanje *Spark* posla na *Dataproc* klasteru i čuvanje rezultata u *Cloud Storage* ili validiranje tabele u *BigQuery* skladištu.

Glavne funkcionalnosti i koristi *Cloud Composer* servisa su [16]:

- **Multiklaud** - Omogućava kreiranje procesa koji povezuju servise iz različitih klaud okruženja, što rezultuje u jedinstveno okruženje za upravljanje tokovima podataka.
- **Otvoreni kod** - Napravljen na osnovu *Apache Airflow*, softvera otvorenog koda.
- **Integracija** - Već gotova integracija sa mnogim *Google Cloud* servisima poput: *Cloud Dataflow, Cloud Dataproc, Cloud Storage, Cloud Pub/Sub* i drugi.
- **Python programski jezik** - Konfiguracija grafova se pravi kroz *Python* programski jezik, što omogućava ponovnu upotrebljivost i modularnost koda, kao i korišćenje raznih biblioteka.
- **Samoodrživost** - Prilikom kreiranja *Cloud Composer* okruženja, korisnici biraju broj instanci koje će činiti klaster, kao i verziju *Python*-a koju žele, a na provajderu je da kreira takvo okruženje sa izabranim brojem instanci i instaliranim *Airflow* softverom. Na ovaj način, krajnji korisnici se mogu fokusirati na održavanje tokova i procesa rada umesto na održavanje infrastrukture i softvera.

Cloud Composer se naplaćuje na nivou jednog minuta u zavisnosti od broja i veličine veb servera, kapaciteta za skladištenje baze podataka koju koristi Airflow i količine odlaznog mrežnog saobraćaja.

4.2 Rudarenje i istraživanje podataka

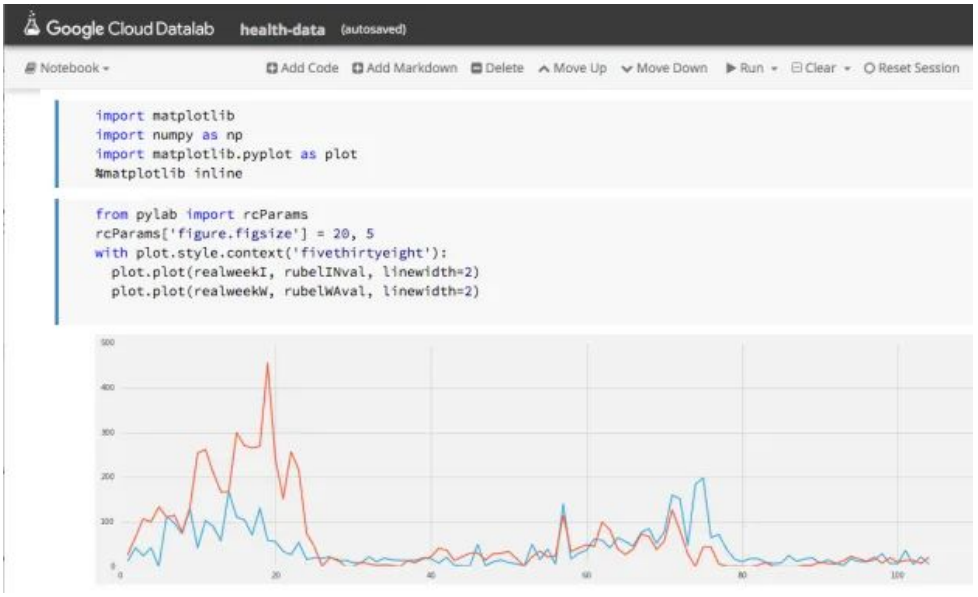
Jedan od glavnih razloga za uvođenje jezera podataka jeste činjenica da nije uvek moguće oblikovati ulazne podatke prema definisanoj šemi i strukturi [17]. Čuvanjem podataka u izvornom obliku u jezeru podataka omogućava nam da odmah sačuvamo sve podatke, a da analizu ostavimo za kasnije. U zavisnosti od prirode sirovih podataka i tipa analitike, procesi i tokovi rada mogu da variraju prema kompleksnosti [17].

S obzirom da većina podataka sačuvanih u jezeru nije odmah spremna za konzumiranje i upotrebu, prvo što je potrebno uraditi jeste da izvršimo određene procedure rudarenje i istraživanja podataka kako bi uvideli potencijalni značaj tog skupa [17]. *Jupyter* je jedan od najčešće korišćenih alata za istraživanje sirovih podataka. *Jupyter* je softver otvorenog koda koji omogućava inženjerima interaktivno izvršavanje koda u različitim programskim jezima kao što su *Python*, R i SQL [32]. Programski kod se piše u radnim sveskama (*notebook*) koje se izvršavaju u veb pretraživaču jer je *Jupyter* veb aplikacija [32].

Google Cloud u svojoj ponudi ima *Cloud Datalab* servis koji predstavlja samoodrživu verziju *Jupyter Notebook-a* [33]. Ovaj servis dolazi sa već instaliranim popularnim softverom za rad sa podacima kao što su *TensorFlow* i *NumPy* [33]. Osim *Datalab* servisa, za potrebe istraživanja sirovih podataka u *Google* ekosistemu su nam dostupni i tradicionalni *Hadoop* alati preko *Dataproc* servisa. Za istraživanje uz pomoć SQL jezika, korisnici mogu sirove podatke da

obrade i srede preko *Dataprep* servisa i učitaju u *BigQuery* skladište podataka. *Dataprep* je servis koji omogućava čišćenje, pripremu i obradu podataka kroz grafički interfejs [27]. Na ovaj način, poslovni korisnici, koji nužno ne poseduju napredno tehnološko znanje, mogu da istražuju određeni sirovi skup iz jezera podataka.

Nakon što se razume analitički potencijal nekog sirovog skupa podataka iz jezera, tada počinje kreiranje procesa i tokova rada koji će taj sirovi skup transformisati i premestiti u neko drugo skladište gde će biti dostupni za analizu širem skupu internih ili eksternih korisnika [17].



Slika 11. Primer *Jupyter* radne sveske u okviru *Cloud Datalab* servisa [33]

4.3 Skladištenje podataka i poslovna inteligencija

Jedna od najbitnijih stvari koju poseduje neka organizacije jesu njeni podaci [20]. Oni se obično koriste za dve svrhe: podršku poslovanju i analitičko donošenje odluka. Jednostavnije rečeno, sistemi za podršku poslovanju su mesto gde podaci ulaze u sistem, a skladište podataka je mesto gde podaci završavaju i postaju spremni za upotrebu [20].

Cilj poslovne inteligencije jeste da omogući pristup podacima svim članovima organizacije kojima su oni potrebni za analitičko donošenje odluka, vođeno činjenicama i podacima [20]. Poslovnim korisnicima treba da se omogući istraživanje podataka uz pomoć ad hok alata poslovne inteligencije. Ovi alati prate paradigmu direktne manipulacije, gde korisnici ne moraju da napišu kod za upite, već oni biraju skup tabela i kolona koje ih zanimaju i dolaze do podataka uz pomoć jednostavnog prevlačenja, duplog klika ili opcije umetanja [19]. Poslovna inteligencija se koristi za poslovnu analitiku, zajedno sa ostalim tehnologijama kao što su prediktivna analitika, mašinsko učenje ili operativno istraživanje softvera [19]. Jedan od najčešće korišćenih rezultata primene poslovne inteligencije jeste dnevni izveštaj sa graficima i tabelama koji obično bude dostavljen na mejl članovima organizacije, i omogućava uvid u trenutno stanje proizvoda na jednostavan način.

Najbitniji zahtevi koje sistemi za skladištenje podataka i poslovnu inteligenciju treba da ispunjavaju su [20]:

- Informacije treba da budu lako dostupne širom organizacije.
- Informacije treba da budu dostavljane često i da izveštaji budu konzistentni.
- Informacije treba da budu dostavljene na vreme.
- Sistem za skladištenje i poslovnu inteligenciju mora da bude prilagodljiv, kao odgovor na rapidne interne i eksterne promene.
- Sistem za skladištenje i poslovnu inteligenciju mora služiti kao autoritativan i poverljiv izvor informacija, koji se koristi za donošenje odluka.
- Poslovni korisnici moraju da prihvate sistem za skladištenje podataka i poslovnu inteligenciju kako bi on postao uspešan.

U ovom potpoglavlju ćemo obraditi arhitekturu dela sistema koji organizacijama može da omogući da od sirovih podataka dođu do adekvatno transformisanih i uređenih podataka u skladištu, spremnih za analizu i dalju upotrebu.

4.3.1 Transformacija podataka

Transformacija podataka je proces u kojem se menja (transformiše) struktura postojećih podataka. Glavni oblici transformacije podataka su [2]:

- **Čišćenje** - Ispravljanje, brisanje ili izmena netačnih ili pogrešnih redova.
- **Agregacija** - Agregiranje podataka kako bi se dobila sumarna verzija datog skupa. Na primer, računanje broja transakcija grupisanih po različitim regionima i kategorijama.
- **Izračunavanje** - Izračunavanje novih vrednosti i metrika na osnovu poslovnih formula, i pretvaranje sirovih podataka u poslovne metrike.

Sirovi podaci, koji se dovlače u platformu za analitiku, su najčešće u transakcionom obliku, što znači da jedan zapis predstavlja jednu poslovnu transakciju [2]. Da bi se ovi podaci koristili za analitiku, potrebno ih je transformisati u pogodniji oblik. Proces transformacije sirovih transakcionih podataka u uređeni skup, pogodan za analitiku, zove se modelovanje podataka [2].

Glavni razlozi zašto je organizacijama potrebna transformacija podataka [2]:

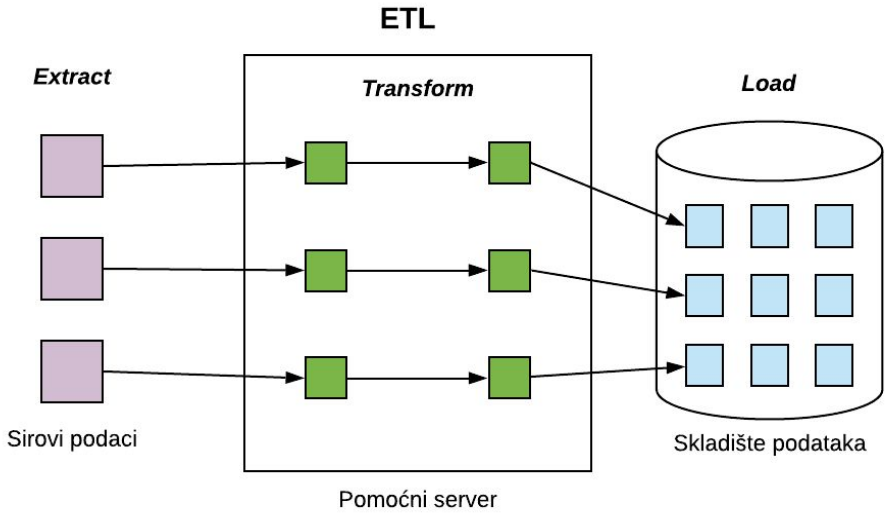
- **Ponovna upotrebljivost** - Svaki korak transformacije podataka može se zamisliti kao zasebna komponenta koja poseduje određenu poslovnu logiku i može se primeniti na više mesta u različitim upitima i izveštajima.
- **Konzistentnost izveštaja** - Zbog navedene ponovne upotrebljivosti, transformacije se pišu samo na jednom mestu i mogu se ponovo upotrebljavati. Ovo dovodi do veće konzistentnosti različitih izveštaja. Na ovaj način se sprečava situacija da dva izveštaja prikažu različite brojeve za istu metriku, zbog načina na koji vrše transformaciju.
- **Bolje performanse** - Procesi transformacije i agregacije podataka se izvršavaju samo jednom za određeni izvorni skup, a procesi izveštavanja i analize nad transformisanim podacima mogu se izvršavati više puta. Sama činjenica da je taj skup očišćen, agregiran i spreman za analitiku daje veliko poboljšanje u performansama.
- **Ekonomska isplativost** - Činjenica da je transformisani skup uređen, znači da će završni upiti i izveštaji koristiti manje procesorske snage, što implicira i manje troškove.

Dve najčešće upotrebljivane paradigme za transformaciju podataka su ETL (*Extract, Transform and Load*) i ELT (*Extract, Load and Transform*) [2].

4.3.1.1 ETL proces

U većini organizacija koje imaju platformu za analitiku, najintenzivniji korak jeste priprema podataka, što podrazumeva kombinovanje, čišćenje i kreiranje skupova podataka spremih za prezentovanje krajnjim korisnicima [2]. Ovaj proces se naziva ETL (*Extract, Transform and Load*). U ETL procesu, izvlače se podaci iz izvornih sistema, zatim se vrše različite transformacije nad tim skupom da bi se na kraju uređeni i transformisani skupovi učitali u skladište podataka. Tri koraka ETL procesa su [2]:

1. **Izvlačenje** (*extract*) podataka iz izvora (pisanje SQL upita koji se izvršava u produkcionim bazama ili dovlačenje JSON, XML ili CSV fajlova).
2. **Transformisanje** (*transform*) podataka u memoriji ETL servera.
3. **Učitavanje** (*load*) transformisanih podataka u skladište.



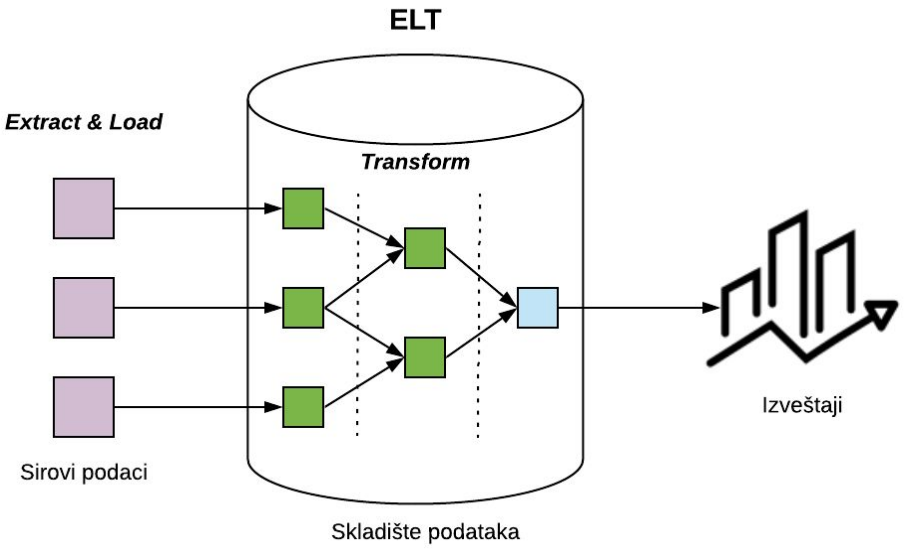
Slika 12. ETL proces [2]

Ključna osobina ETL procesa jeste da se sirovi podaci transformišu izvan skladišta podataka, obično uz pomoć stejdžing (*staging*) servera i na kraju se samo transformisani podaci učitavaju u skladište [2].

4.3.1.1 ELT proces

ELT (*Extract, Load and Transform*) proces koristi drugačiju paradigmu prilikom transformacije podataka. Centralna komponenta ELT procesa jeste skladište podataka [2]. Umesto da se podaci transformišu pre nego što dođu u skladište, prvo se učitavaju sirovi podaci u skladište da bi se onda u okviru skladišta vršile dalje transformacije. Tri koraka ELT procesa su [2]:

1. **Izvlačenje** (*extract*) podataka iz izvora.
2. **Učitavanje** (*load*) sirovih podataka u skladište.
3. **Transformisanje** (*transform*) podataka unutar skladišta podataka.



Slika 13. ELT proces [2]

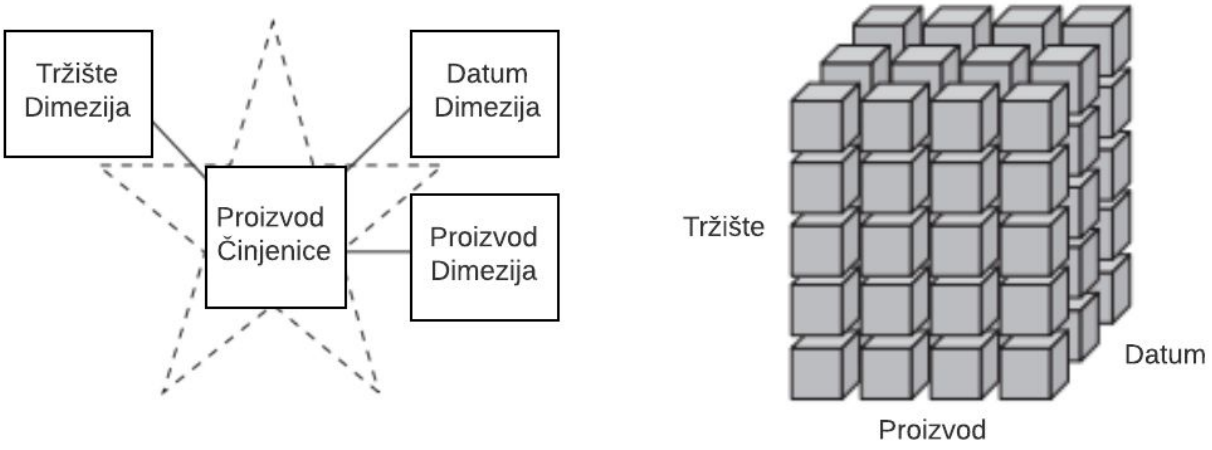
4.3.2 Dimenziono modelovanje

Dimenziono modelovanje je tehnika modelovanja podataka, korišćena za izradu poslovnih aplikacija koje omogućavaju intenzivne analitičke upite kroz visoke performanse upita i upotrebljivost podataka koji se analiziraju [19]. Analitički upiti koji imaju koristi od dimenzionog modela su upiti koji analiziraju poslovne podatke i koji često izvode akcije kao što su sumiranje, računanje proseka ili prikazivanje trendova [19]. Dimenziono modelovanje je gotovo svuda prihvaćena tehnika za predstavljanje analitičkih podataka zbog toga što istovremeno ispunjava dva veoma bitna zahteva [20]:

1. Dostaviti podatke koji su razumljivi poslovnim korisnicima.
2. Obezbediti brzo izvršavanje upita.

Relacione baze podataka, koje služe kao operativna podrška aplikacijama i sistemima, obično su organizovane i napravljene tako da predstavljaju normalizovani model. To je model u kojem su podaci podeljeni na pojedinačne entitete koji su zatim sačuvani u zasebne tabele, a ti entiteti su međusobno povezani preko stranog ključa [20]. Normalizovani model kao posledicu ima sprečavanje pojave dupliciranih podataka i vrednosti. Ovaj model, iako veoma dobar za potrebe podrške poslovanju, postaje previše kompleksan i neupotrebljiv za potrebe poslovne inteligencije i ad hok izveštavanja [20].

Dimenzioni model sadrži iste informacije kao i normalizovani model, ali on ih pakuje u drugačiji format, koji kao rezultat donosi razumljivost, bolje performanse upita i otpornost na promene [20]. Dimenzioni model, implementiran u relacionim bazama podataka, se zove “zvezda šema”, zbog sličnosti njegove strukture sa oblikom zvezde. Dimenzioni model, implementiran u multidiomenzionom okruženju baza podataka, se zove OLAP (*online analytical processing*) kocka [20]. Ova dva načina implementacije dimenzionog modela imaju velike sličnosti u pogledu logičkog dizajna, ali fizička implementacija se dosta razlikuje.



Slika 14. Zvezda šema nasuprot OLAP kocke [20]

Iako se u poslednje vreme mogućnosti OLAP kocki znatno poboljšavaju, preporučuje se da se detaljne i atomične informacije prvo učitaju u zvezda šemu, i onda opciono odatle da se popune OLAP kocke [20]. Struktura zvezda šeme se sastoji od dve vrste tabela: tabele činjenica i tabele dimenzija.

Tabela činjenica u dimenzionom modelu je mesto u kojem se čuvaju merenja učinka koja su rezultat poslovnih procesa [20]. Dobra je praksa da se merenja niskog nivoa, nastala od jednog poslovnog procesa, sačuvaju u jednom dimenzionom modelu. S obzirom da su podaci merenja ubedljivo najveći skup podataka jedne organizacije, oni ne bi trebali da budu replicirani na više mesta. Omogućavanje poslovnim korisnicima širom organizacije da pristupe centralnom skladištu za svaki skup metrika obezbeđuje konzistentnost podataka kroz celu organizaciju. Termin činjenica (*fact*) predstavlja jednu poslovnu meru ili metriku, a svaki red u tabeli činjenica odgovara jednom poslovnom događaju (transakcija, prodaja proizvoda, zakazivanje hotela i slično). Ideja da, jedan merljivi događaj u fizičkom svetu ima jedan na jedan odnos sa odgovarajućim redom u tabeli činjenica, predstavlja osnovni princip dimenzionog modelovanja i sve ostalo se gradi na ovoj osnovi [20].

Tabele dimenzija su nezaobilazni pratioci tabele činjenica [20]. Ove tabele sadrže tekstualni kontekst u vezi sa merljivim događajima jednog poslovnog procesa. One odgovaraju na pitanja “Ko, šta, gde, kada, kako i zašto?” za jedan poslovni događaj. Tabele dimenzija obično imaju veći broj kolona ili atributa, i nije neobično da taj broj bude i trocifren [20]. Za razliku od tabela činjenica koje imaju veliki broj redova, tabele dimenzija obično imaju manji broj redova ali veći broj tekstualnih kolona [20]. Svaki red (dimenzija) u ovim tabelama je jedinstveno određen sa primarnim ključem koji predstavlja osnovu za referencijalni integritet sa bilo kojom tabelom činjenica sa kojom se radi *join*.

Da bi od sirovih podataka došli do uređenog dimenzionog modela u skladištu podataka, neophodno je uspostaviti adekvatan proces transformacije u našoj platformi za analitiku. U sledećim potpoglavljima ćemo detaljnije razmotriti koje su neke od opcija za transformisanje podataka, sa konačnim ciljem kreiranja dimenzionog modela, pogodnog za poslovnu inteligenciju. S obzirom da su primeri arhitekture usmereni na *Google Cloud* platformu, za skladište podataka je odabran servis *BigQuery*.

4.3.2.1 Transformacija van skladišta

Transformacija podataka van skladišta i učitavanje samo očišćenih, transformisanih i uređenih podataka u skladište jeste proces koji prati ranije spomenutu ETL paradigmu, gde se podaci prvo ekstraktuju, zatim transformišu i na kraju učitaju u krajnju destinaciju. Razmotrimo detaljnije tri ključna koraka u ETL procesu specifična za proces kreiranja dimenzionog modela u skladištu podataka:

1. **Izvlačenje podataka iz izvora** - U slučaju naše platforme za analitike, većina sirovih podataka se već nalazi u jezeru (*Cloud Storage*) koji postaje glavni izvor, dok drugi deo podataka mogu da čine strimovani podaci, koje možemo direktno da čitamo iz servisa za razmenu poruka kao što je *Cloud Pub/Sub,* tako da podaci o događajima mogu da preskoče jezero da bi se što ranije pojavili u skladištu podataka i bili dostupni za analizu krajnjim korisnicima.
2. **Transformisanje podataka** - Ovo je ključni korak u procesu kreiranja dimenzionog modela, jer se ovde izvršava čišćenje i priprema podataka iz različitih izvora, kako bi se oni međusobno spojili (*join*), praveći skupove podataka koji će eventualno završiti u tabelama dimenzija i činjenica. Korak transformacije se zapravo sastoji iz većeg broja manjih koraka obrade koji se mogu izvršvati uz pomoć više različitih servisa za obradu podataka u oblaku.
3. **Učitavanje podataka u skladište** - Samo obrađeni i filtrirani podaci se na kraju procesa učitavaju u skladište podataka, popunjavajući tabele dimenzija i činjenica. Na ovaj način se u skladištu podataka nalaze samo obrađeni podaci, predstavljeni preko dimenzionog modela, a sirovi podaci se nalaze u jezeru podataka.

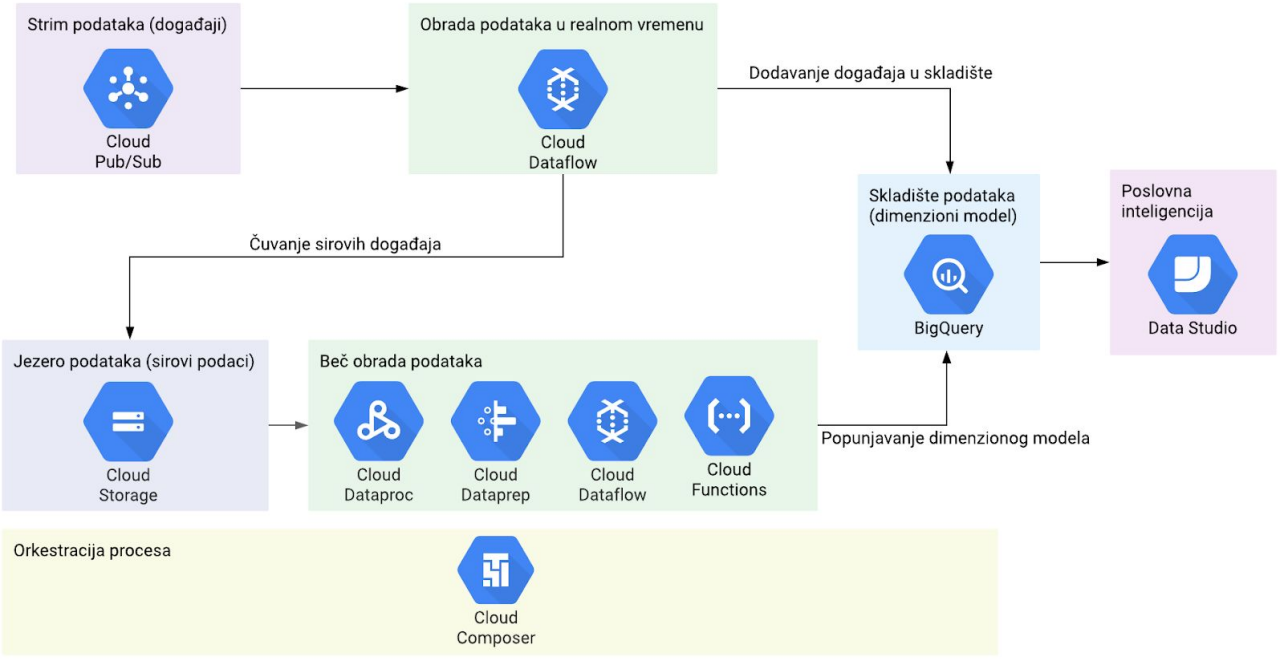
S obzirom na veliki broj raznovrsnih izvora iz kojih podaci mogu da dolaze, kao i na samu količinu i brzinu tih podataka, ali i međusobnu zavisnost transformacionih koraka, ovaj proces može da postane veoma kompleksan. Proces u kojem podaci protiču kroz različite delove sistema, od izvora do destinacije, menjajući svoju strukturu i format, se često naziva pajplajn podataka (*data pipeline*) [1]. Pajplajni podataka treba da budu konfigurisani: podešavaju se tačna vremena ponovnog izvršavanja (svaki dan, svaki sat), određuju se međuzavisnosti koraka u pajplajnu, kako bi se znao tačan redosled kojim se koraci smeju izvršavati. Postoje dva osnovna pristupa u kreiranju i konfiguraciji pajplajna podataka: konfiguracija kroz kod i konfiguracija kroz grafički interfejs.

4.3.2.1.1 Konfiguracija pajplajna kroz kod

U ovom pristupu se različiti koraci ETL procesa definišu kroz kod u nekom programskom jeziku. Jedan od najpopularnijih alata za ove potrebe jeste ranije pomenuti *Apache Airflow*, gde se celokupni proces definiše kroz Python skriptu [31]. U našoj arhitekturi, za konfiguraciju pajplajna i orkestraciju svih koraka ćemo koristiti *Cloud Composer, Google Cloud* samoodrživu verziju *Airflow* softvera. Ovaj pristup je tehnički zahtevniji jer zahteva konfigurisanje integracije između različitih servisa, ali sa druge strane, dobijamo mogućnost prilagođavanja. Za konstrukciju ovog sistema neophodno je iskoristiti servise koji pripadaju različitim slojevima platforme za analitiku:

- ***Cloud Storage*** - Njegova osnovna svrha u ovom delu sistema jeste da služi kao izvor sirovih podataka, tj. za fizičku implementaciju jezera podataka. Sirovi podaci prolaze kroz niz transformacionih koraka koji se izvršavaju uz pomoć drugih klaud servisa, a potencijalno se *Cloud Storage* koristi i kao mesto za skladištenje međurezultata obrade.

- ***Cloud Dataproc*** - Servis koji nudi održavanje *Hadoop* klastera sa velikim izborom alata karakterističnih za ovo okruženje, kao što su *Spark, Hive, Presto* i drugi [26]. *Dataproc* se koristi kao jedan od glavnih servisa za obradu podataka u pajplajnima iz razloga što alati poput *Spark* omogućavaju paralelnu obradu velikih skupova podataka na većem broju mašina u klasteru i na taj način čini velike skupove podataka obradivim u relativno kratkom vremenskom periodu. *Dataproc* ima u sebi i HDFS (*Hadoop Distributed File System*) komponentu za skladištenje podataka, ali se u danas obično odvajaju resursi obrade i resursi skladištenja. Prema ovoj paradigmi se *Dataproc* klaster koristi samo za obradu podataka, a međurezultati i krajnji izlaz se čuvaju obično u *Cloud Storage* ili nekom drugom servisu za skladištenje [26].
- ***Cloud Dataprep*** - Servis koji omogućava kreiranje transformacionih procesa za čišćenje, obradu i pripremu podataka kroz grafički interfejs i širok izbor postojećih koraka obrade (tzv. recepata) [27]. Može se uključiti u pajplajn tako što čita podatke sa nekog izvora kao što je *Cloud Storage*, obradi ih i upiše nazad u neki servis za čuvanje podataka [27].
- ***Cloud Dataflow*** - Samoodrživ servis koji omogućava kreiranje pajplajna sa beč obradom kao i sa obradom u realnom vremenu [24]. *Google* je preuzeo odgovornost za održavanje infrastrukture za ovaj servis, a krajnji korisnici se mogu fokusirati na kreiranje procesa i tokova rada koji čine pajplajn podataka. *Cloud Dataflow* koristi softver otvorenog koda *Apache Beam* kao distribuirani sistem za obradu podataka [24]. Koraci za obradu podataka se pišu u programskom jeziku *Java* koristeći *Apache Beam* frejmvork. U našem sistemu *Dataflow* će takođe biti integrisan sa striming platformom *Cloud Pub/Sub,* odakle će dovlačiti događaje u realnom vremenu, zatim obraditi, i u veoma kratkom vremenskom roku prebaciti ih u skladište podataka.
- ***Cloud Functions*** - Samoodrživ servis koji omogućava pisanje i izvršavanje funkcija u nekoliko standardnih programskih jezika, bez upravljanja infrastrukture na kojoj se taj kod izvršava [28]. Dozvoljeno je pisanje funkcija u programskim jezicima *Python*, *Java*, *Go* i *Node.js*. Korisnici plaćaju ovaj servis proporcionalno vremenu izvršavanje koje je bilo potrebno njihovim funkcijama. U pajplajnu podataka se ovaj servis obično koristi za neke jednostavnije i kraće zadatke zbog limita izvršavanja od 9 minuta [28] i činjenice da se taj kod izvršava na jednoj mašini. Često se koristi za orkestraciju poslova i prosleđivanje podataka između servisa.
- ***BigQuery*** - Samoodrživ servis za skladištenje podataka i masivno paralelizovanu obradu podataka (*massive parallel processing*) [15]. *BigQuery* će predstavljati krajnju destinaciju našeg pajplajna i mesto u kojem će biti kreiran dimenzioni model podataka. Sa njim će se dalje integrisati alati za poslovnu inteligenciju, koji će omogućiti interaktivni prikaz podataka iz skladišta.
- ***Cloud Composer*** - Servis za orkestraciju zasnovan na *Apache Airflow* softveru, koji će biti glavni koordinator svih ostalih servisa i mesto u kojem ćemo kroz *Python* kod konfigurisati celokupni pajplajn podataka predstavljen preko usmerenog acikličnog grafa [16]. *Cloud Composer* je zadužen za zakazivanje i redovno izvršavanje celog pajplajna, jer podaci obično pristižu svakodnevno u izvore, i neophodno je celi proces izvršavati iznova da bi u skladištu imali sveže podatke.
- ***Data Studio*** - Alat za interaktivnu poslovnu inteligenciju sa velikim brojem integracija sa ostalim servisima iz *Google Cloud* ponude [29]. Primarno će prikazivati podatke iz *BigQuery* skladišta. Omogućava kreiranje interaktivnih izveštaja.



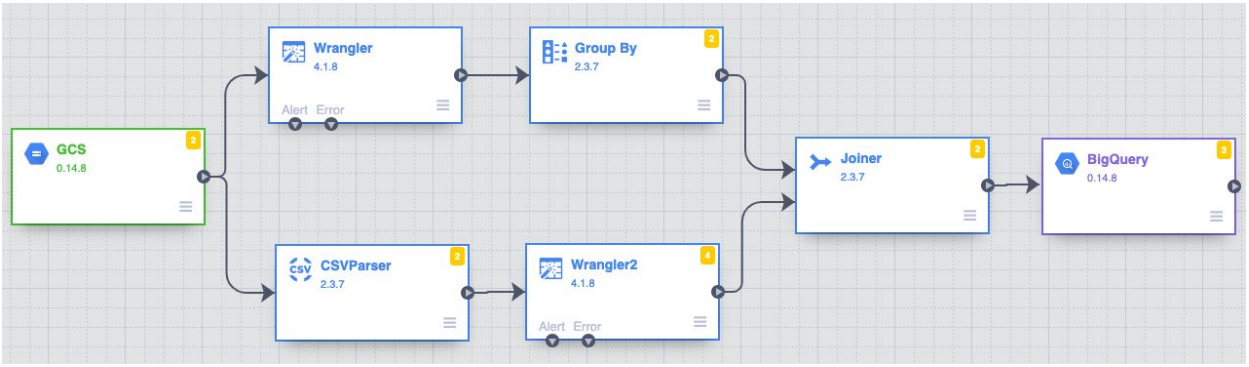
Slika 15. Arhitektura ETL procesa za kreiranje dimenzionog modela

4.3.2.1.2 Konfiguracija pajplajna kroz interfejs

Drugi popularan način za kreiranje pajplajna za protok podataka kroz sistem jeste kroz upotrebu servisa koji nudi grafički interfejs za integraciju različitih servisa i kreiranje kompletnog grafa međuzavisnosti. Ovaj pristup se često naziva pajplajni bez koda, jer za razliku od pravljenja pajplajna uz pomoć alata kao što je *Airflow* gde se sva konfiguracija definiše kroz kod, u ovom pristupu znanje programskog jezika nije neophodno i sve se radi kroz interfejs.

Google Cloud u svojoj ponudi ima servis *Cloud Data Fusion,* koji omogućava integraciju različitih servisa, kreiranje i upravljanje pajplajnima podataka, zasnovan na softveru otvorenog koda CDAP (*Complex Data Pipelines*) [21]. Ovaj servis nudi grafički interfejs, koji za cilj ima da poveća efikasnost kreiranja pajplajna podataka i da smanji kompleksnost kreiranje pajplajna. Glavna ideja ovakvih servisa jeste da automatizuju posao kreiranja pajplajna podataka i da neke delove posla, za koje je inače potreban inženjer obrade podataka, preuzme i završi sam servis. Uz pomoć servisa ovog tipa, pravljenje sistema za integraciju podataka nije više ograničeno samo na inženjere, već i korisnici sa manje tehnološkog iskustva mogu da naprave svoje procese ekstrakcije, transformacije i učitavanja podataka, kako bi na kraju obrađeni podaci završili u skladištu i bili spremni za prikaz u nekom od alata za poslovnu inteligenciju [21].

Paradigma konfigurisanja pajplajna kroz kod je zastupljenija u tehnološkim startap kompanijama, gde je mogućnost promene i prilagođavanja specifičnim zahtevima veoma poželjna, kao i ekonomičnost samog rešenja [34]. Paradigma konfigurisanja pajplajna kroz interfejs je više zastupljena u enterprajz (*enterprise*) svetu, gde kompanije koje možda i nisu primarno tehnološke i nemaju adekvatan kadar, dobijaju mogućnost lakog kreiranja pajplajna bez pisanja koda, na račun skuplje cene i manje prilagodljivosti [34].



Slika 16. Primer pajplajna kreiranog uz pomoć *Cloud Data Fusion* servisa

4.3.2.2 Transformacija unutar skladišta

Istorijski je izgradnja skladišta podataka bila veoma skup proces, kako sa softverskog aspekta tako i sa hardverskog [2]. Cena servera, implementacije i softverskih licenci za jedan projekat skladištenja podataka pre 20 ili 30 godina je neretko dostizala i milione dolara, a obično su bili potrebni meseci ili godine za implementaciju [2]. S ozbirom na potencijalno veoma visoku cenu, kompanije su morale da vode računa o troškovima pa su zbog toga u skladište učitavali samo očišćene, adekvatno transformisane i agregirane podatke. U to vreme je u svetu softverskog inženjerstva bio i dalje dominantan princip vodopada pa je bilo prihvatljivo da se na početku projekta izdvoji vreme da se isplaniraju neophodne transformacije podataka koje bi se izvršile pre nego što podaci završe u skladištu [2]. Sa ovog stanovištva, ETL proces, u kojem su se samo obrađeni podaci učitavali u skladište podataka, je imao najviše smisla.

Danas su se neke stvari znatno izmenile u tehnološkom svetu i ELT proces u kojem se u skladšte podataka učitavaju podaci koji nisu u potpunosti obrađeni postaje sve popularniji [2]. Promene koje su imale veliki uticaj na popularizaciju ELT procesa su [2]:

- **Dostupnost i ekonomičnost skladišta podataka u oblaku** - Moderna skladišta podataka danas su u stanju da čuvaju i obrađuju veliku količinu podataka za relativno malu cenu.
- **Eksplozija u količini i raznovrsnosti sakupljenih podataka** - Revolucija velikih podataka je kao razultat imala veliki razvoj novih alata i tehnologija sposobnih za obradu skupova podataka reda veličine nekoliko terabajta.
- **Popularizacija agilnog razvoja softvera** - Ovo je značilo da su timovi zaduženi za analitiku morali da prate agilne principe razvoja, da budu prilagodljivi i spremni na brze iteracije.

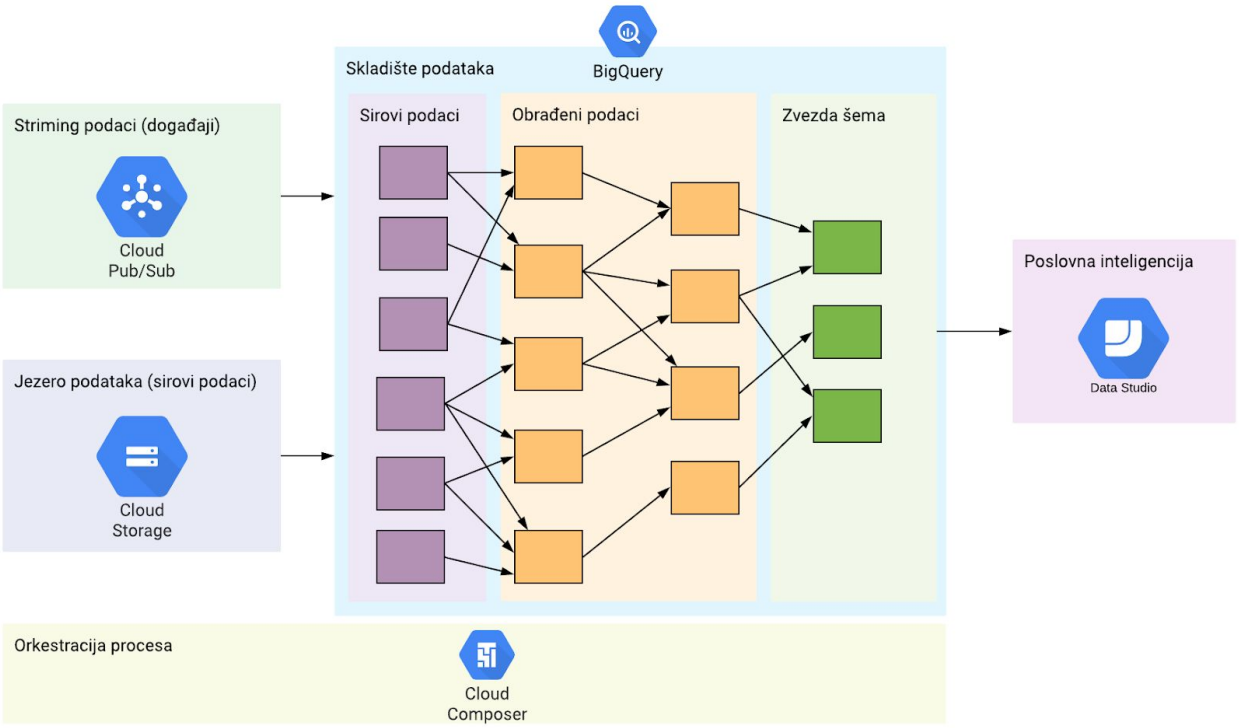
Slično kao i kod transformacija van skladišta, ukoliko se opredelimo za transformacije unutar skladišta postoje dve opcije za implementaciju tih transformacija i kompletnog modelovanja podataka: modelovanje kroz kod i modelovanje kroz interfejs.

4.3.2.2.1 Modelovanje kroz kod

Proces kreiranja sistema za modelovanje podataka unutar skladišta je veoma sličan procesu kreiranja ETL sistema za transformisanje podataka van skladišta. I dalje će se koristiti neki softver za orkestraciju transformacija, u našem slučaju to je *Cloud Composer*, a razlika je u servisima koji će izvršavati konkretne transformacije, agregacije, deduplikacije i ostale manipulacije nad podacima. U ETL sistemu, *Cloud Composer* je bio glavni medijator između

većeg broja različitih servisa za obradu i bio je zadužen za njihovu integraciju. U ELT sistemu se sva obrada podataka vrši unutar skladišta, što znači da je u našem slučaju *BigQuery*, moderno klaud skladište podataka, jedini servis koji će biti zadužen za obradu podataka. Pojedinačni koraci u ovom procesu su zapravo transformacije nad tabelama u skladištu, tačnije to su SQL upiti koji se orkestriraju preko *Cloud Composer*-a, a izvršavaju uz pomoć *BigQuery* servera za obradu. Zasebni SQL upiti koji predstavljaju korake obrade imaju određene međuzavisnosti koje su predstavljene direktnim acikličnim grafom definisanim u *Python* skripti, na osnovu koje *Cloud Composer* zna kojim redosledom se transformacije trebaju izvršavati.

ELT pristup donekle smanjuje kompleksnost celog pajplajna, jer se sva obrada vrši preko SQL upita unutar samo jednog servisa (*BigQuery*). Ovaj pristup omogućava kreiranje novih modela i transformacija većem broju članova organizacije, jer je SQL postao standard koji je široko rasprostranjen. ELT pristup ima i određene mane, kao što je ranije napomenuto. SQL jezik donosi mnoge prednosti, ali obrada podataka bez pravog programskog jezika sa osnovnim principima modularnosti, ponovne upotrebljivosti i mogućnosti testiranja predstavlja manu ovog pristupa [1].



Slika 17. Arhitektura ELT procesa sa orkestracijom *Cloud Composer*-a

4.3.2.2.2 Sloj modelovanja podataka

Proces mapiranja sirovih podataka u format koji je razumljiv poslovnim korisnicima naziva se modelovanje podataka [2]. Osim razumljivosti podataka, postoje i drugi razlozi za modelovanje kao što su performanse i mogućnost istraživanja podataka, ali u najosnovnijem obliku, modelovanje podataka predstavlja pretvaranje sirovih skupova u korisne poslovne metrike [2].

Sloj modelovanja podataka je sistem koji sadrži mapiranje između poslovne logike i pravila fizičkog skladištenja podataka. Primarno postoji u ELT paradigmi, gde se podaci učitavaju u skladište pre nego što budu transformisani. U kontekstu sloja modelovanja podataka, modelovanje zapravo predstavlja proces izgradnje i održavanja ovog sloja.

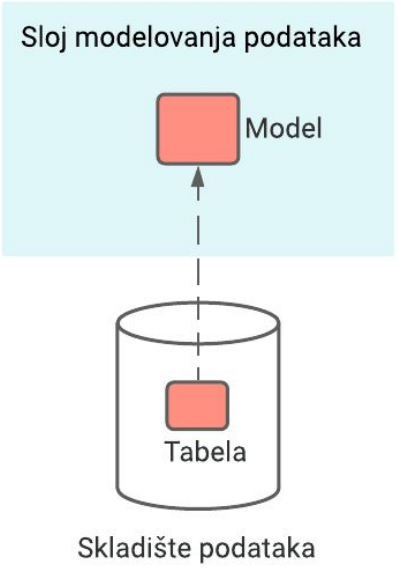
Sloj modelovanja podataka se obično povezuje sa nekim alatima za vizuelizaciju ili poslovnu inteligenciju. Korisnici sa manjim tehnološkim iskustvom mogu da pristupe takvim alatima, interaguju sa korisničkim interfejsom i dobiju analitiku koja im je potrebna, bez potrebe da pričaju sa nekim inženjerom ili tehnološkim članom organizacije. Sa adekvatno napravljenim i održavanim slojem modelovanja podataka organizacija dobija sledeće koristi [2]:

- Rukovodstvo organizacije može da upotrebi alat za poslovnu inteligenciju da dobije prave odgovore na svoja pitanja bez potrebe da se kontaktiraju analitičari ili inženjeri. Na ovaj način se podstiče samousluživanje poslovnih korisnika informacijama.
- Glavni fokus analitičara i inženjera postaje održavanje i unapređivanje sloja modelovnja podataka i pajplajna podataka bez ranije učestalih ad hok eksternih zahteva.
- Cela organizacija poseduje dobro dokumentovan sloj sa znanjem o podacima i načinima na koji su ti podaci modelovani i sačuvani. Na ovaj način, ako određeni analitičari ili inženjeri odluče da napuste kompaniju, znanje o podacima je sačuvano i organizovano u sloju modelovanja podataka.

Danas postoji nekoliko alata za modelovanje podataka od kojih su dva najpoznatija *Holistics* i *Looker*. Ovi alati imaju određene zajedničke karakterstike [2]:

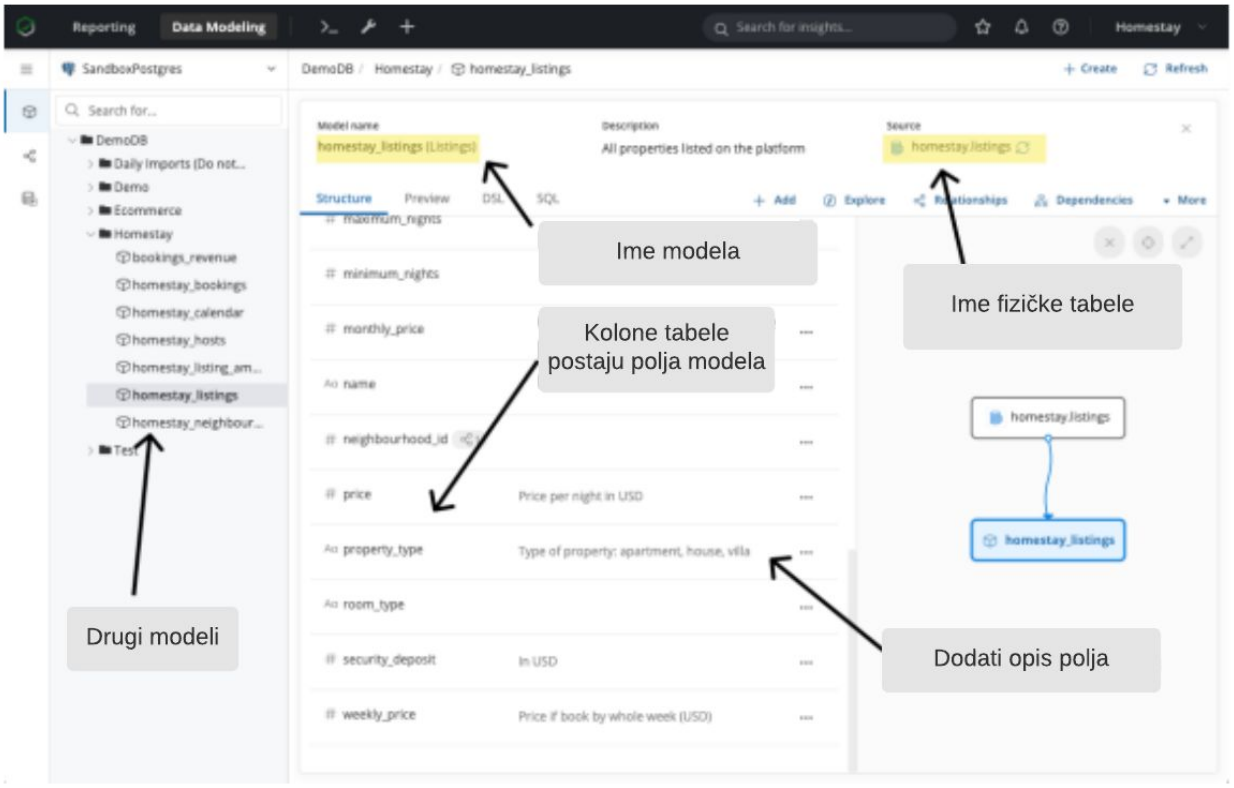
- Povezuju se sa skladištem podataka.
- Posmatraju proces modelovanja podataka kao transformisanje starih tabela u nove tabele.
- Automatski generišu SQL upite koji će transformisati podatke.
- Omogućavaju korisnicima da anotiraju, prate i upravljaju promenom modela kroz vreme.
- Omogućavaju korisnicima da prate celokupni protok podataka kroz sistem u jednom alatu.

Prilikom manipulacije podataka u sloju modelovanja, obično je slučaj da se zapravo ne menja fizička tabela, već se napravi apstraktni objekat koji će biti vezan za tabelu i omogućiti lakšu manipulaciju. Model podataka je apstraktni pogled koji se nalazi iznad fizičke tabele u bazi, koji se može manipulisate bez direktne izmene fizičkih podataka. Većina alata za modelovanje dozvoljavaju dodavanje metapodataka kako bi obogatili model i podatke.



Slika 18. Odnos između modela i fizičke tabele

Alati za modelovanje omogućavaju kreiranje jednostavnih modela koji će biti vezani za jednu fizičku tabelu, to možemo posmatrati kao način predstavljanja fizičke tabele u sloju za modelovanje. Kolone iz fizičke table se mapiraju na polja modela, gde je moguće dodati i nova posebna polja koja su nastala kombinacijom ili agregacijom više kolona iz fizičke tabele. Nakon što napravimo jednostavne modele, ovi alati dozvoljavaju kreiranje novih kompozitnih modela koji će nastati kombinacijom više jednostavnih modela. Na taj način se kreira graf međuzavisnosti slično kao kod kofnigurisanja ETL procesa kroz kod i *Cloud Composer.* Ono što čini ovaj pristup drugačijim jeste što se te međuzavisnosti, načini spajanja, kao i polja koja će se koristiti definišu kroz interfejs alata (npr. Holistsics) ili posebni domenski specifičan jezik karakterističan za taj alat (npr. *LookML* kod *Looker-a*), a sam alat je zadužen za kreiranje celokupnih grafova izvršavanja kao i SQL skripti, koje će se izvršavati nad skladištem podataka. Na ovaj način, modelovanje podataka postaje dostupno široj grupi ljudi unutar organizacije, jer za razliku od konfigurisanja ELT procesa kroz kod gde je potrebno pisanje SQL upita za transformaciju podataka, u ovom pristupu se opiše kako trebaju izgledati novi modeli, a alati za modelovanje na sebe preuzimaju odgovornost kreiranja grafa zavisnosti i SQL transformacija kao i njihovo zakazivanje i izvršavanje u skladištu (npr. *BigQuery*).



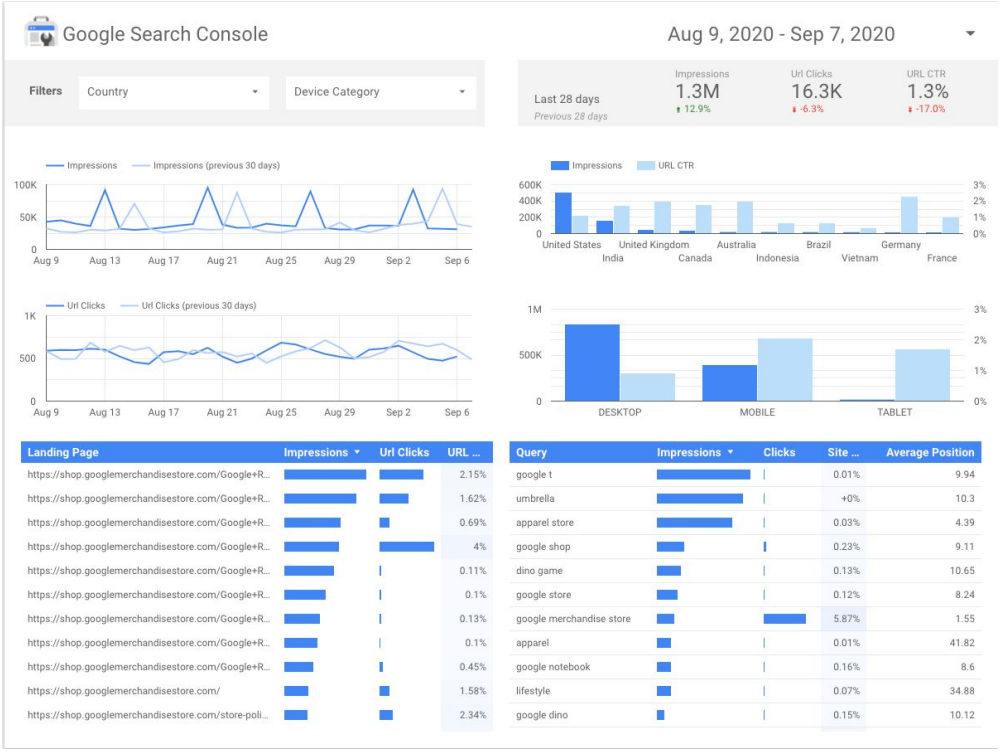
Slika 19. Primer interfejsa Holistics softvera - alata za modelovanje podataka

Alati kao što su *Holistics* i *Looker* osim sloja modelovanja podataka u sebi sadrže i neophodne alate za poslovnu inteligenciju. Potencijal da jedan alat bude zadužen za celokupni proces poslovne inteligencije od transformacije sirovih podataka do interaktivnog i ad hok izveštavanja je glavni razlog ekspanzije ovih alata.

4.3.3 Alati poslovne inteligencije

Alati poslovne inteligencije i aplikacije koje pristupaju podacima iz skladišta moraju biti jednostavni za upotrebu. Rezultati upita trebaju se izračunati sa minimalnim čekanjem kako bi krajnji korisnici došli do željenih rezultata u prihvatljivom vremenu [20]. Tradicionalni alati za poslovnu inteligenciju (npr. *MicroStrategy* i *Sisense*) su imali monolitsku strukturu, što znači da su ti alati dolazili sa ugrađenim skladištem podataka i da se obrada podataka vršila unutar tog servisa. Moderni klaud alati za poslovnu inteligenciju predstavljaju sloj koji se nalazi iznad skladišta podataka gde se zapravo čuvaju podaci. Zadatak ovih alata jeste da korisnički zahtev napravljen kroz manipulaciju korisničkog interfejsa pretvore u SQL upit, koji će se izvršavati unutar samog skladišta (npr. *BigQuery*).

Strukturirani oblici poslovne inteligencije, kao što su izveštaji ili komandne table predstavljaju rekonstituisane podatke na nivou interfejsa [18]. Ovo za posledicu ima da poslovni korisnici treba samo da razumeju strukturu modela podataka koja čini osnovu za izveštavanje i druge vidove poslovne inteligencije koje oni koriste [18]. Obim poslovne inteligencije se raširio od streteških pitanja do operativnih zadataka, pa shodno tome broj zaposlenih koji treba da budu u mogućnosti da je koriste se povećava [19]. Moderni klaud alati za poslovnu inteligenciju su omogućili samostalno istraživanje i izveštavanje širem opsegu članova organizacije.



Slika 20. Primer izveštaja napravljenog uz pomoć *Google Data Studio* servisa

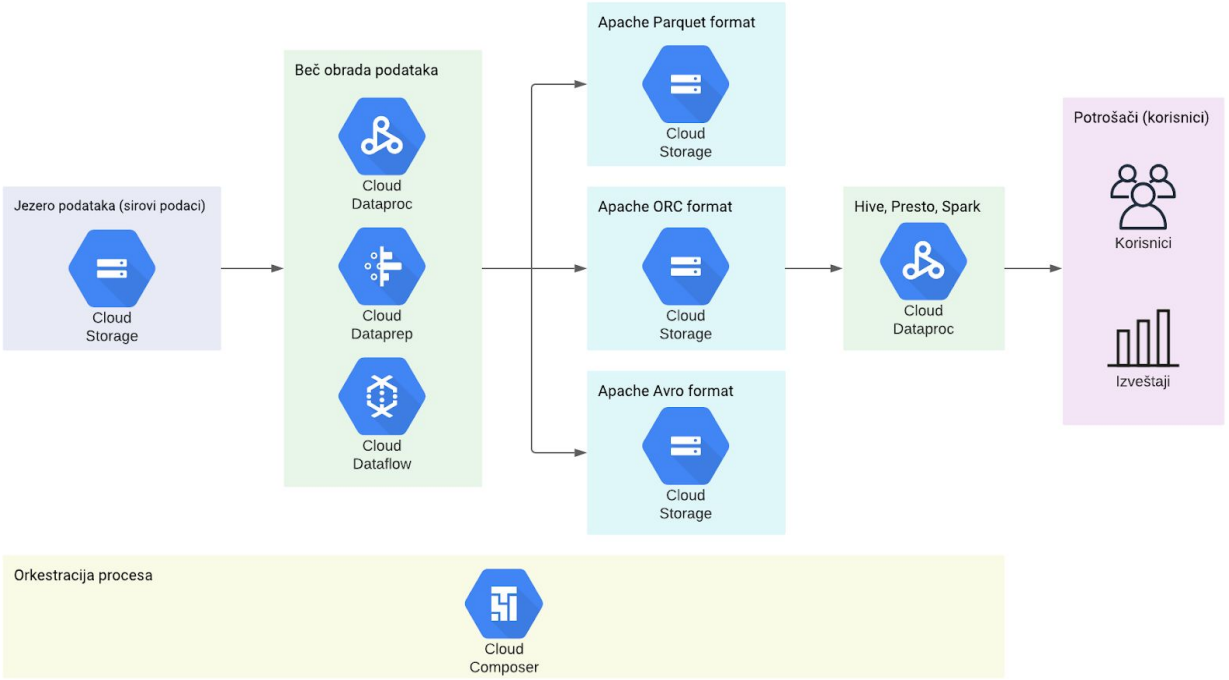
4.4 Beč analitika

S obzirom na tradicionalnu popularnost alata iz *Hadoop* ekosistema kao i razvoj novih tehnologija koje su doprinele njegovom značaju, veliki broj organizacija danas ima potrebu za kreiranjem sistema za izvođenje beč (*batch*) analitike uz pomoć *Hadoop* alata. *Google Cloud* u

svojoj ponudi ima veći broj servisa koji se mogu upotrebiti za implementaciju *Hadoop* platforme u modernom okruženju u oblaku. Za razliku od tradicionalnih *Hadoop* rešenja, gde su podaci bili sačuvani na *Hadoop* klasteru u sklopu HDFS (*Hadoop Distributed File System*), u modernom klaud okruženju dolazi do odvajanja resursa za skladištenje i resursa za obradu. Svi podaci sa kojima rade alati za obradu u ovom sistemu će biti sačuvani na *Cloud Storage* koje predstavlja naše jezero podataka. Na ovaj način smo odvojili skladištenje od obrade i u slučaju da se poveća količina podataka sa kojom naš sistem treba da radi, to će adekvatno da isprati *Cloud Storage,* koji se nezavisno skalira od *Hadoop* klastera, i koji će služiti primarno za obradu podataka.

Još jedna velika prednost koju dobijamo prelaskom u oblak jeste da *Hadoop* klastere sada možemo posmatrati kao kratkotrajne i prolazne resurse, što zapravo znači da kada je korisnicima potreban *Hadoop* klaster određene veličine, oni mogu da u tom trenutku zatraže njegovo kreiranje koje će trajati nekoliko minuta i nakon što se završe neophodni poslovi i rezultati sačuvaju u jezero podataka, taj klaster može da se izgasi. Na ovaj način organizacije ne plaćaju za resurse kada im oni nisu potrebni.

Kao što je ranije spomenuto, *Cloud Dataproc* je servis koji omogućava korisnicima pristup *Hadoop* klasterima na zahtev sa već instaliranim odabranim alatima iz *Hadoop* ekosistema [26]. Alati koji se danas najčešće koriste na *Dataproc* klasterima za beč obradu su *Spark*, *Hive* i *Presto*. Svaki od ovih alata je sposoban za paralelnu i distribuiranu obradu gde se dostupni resursi u klasteru adekvatno mogu upotrebiti za obradu velike količine podataka. Nakon što se završi obrada podatka, rezultati se čuvaju u *Cloud Storage*, obično u nekom od binarnih formata kao što su *Apache Avro*, *Apache Parquet* i *Apache ORC* [17]. Čuvanje podataka u binarnim formatima donosi prednosti performansi prilikom izvršavanja upita nad tim podacima, kao i mogućnost bolje kompresije za smanjenje količine podataka koju je neophodno pročitati iz jezera i preneti preko mreže.



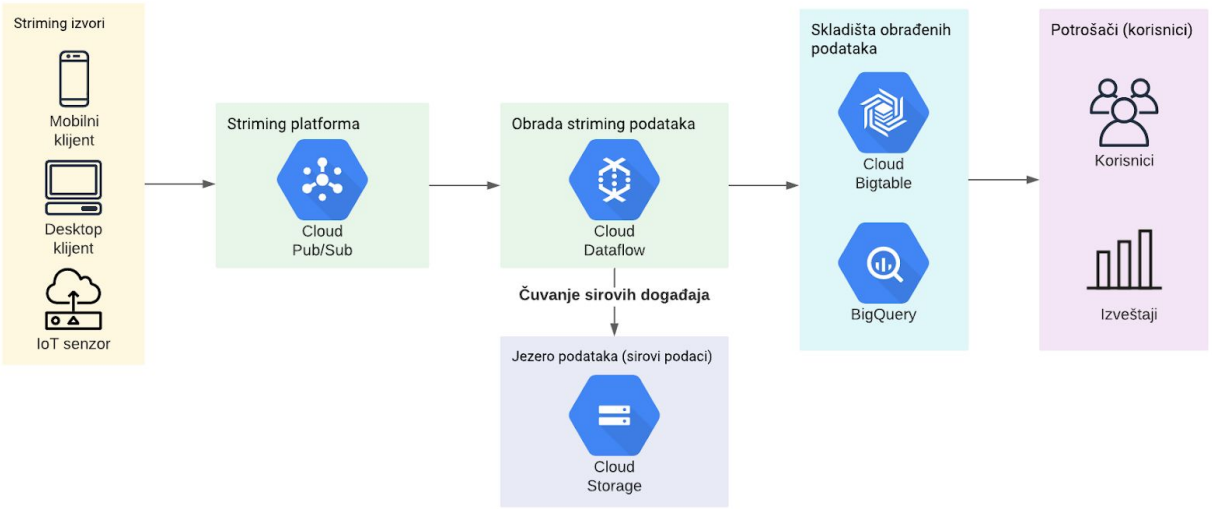
Slika 21. Proces beč analitike na *Google Cloud* platformi

4.5 Analitika u realnom vremenu

Za razliku od beč analitike koja se obično sastoji od dugotrajnih poslova koji se često izvršavaju po noći kako bi generisali dnevne i mesečne izveštaje, cilj analitike u realnom vremenu jeste da se dođe do novih informacija u što kraćem vremenskom periodu. Organizacije se danas nadmeću u tome da maksimalno skrate vremenski interval koji protekne od generisanja strimovanih podataka do njihove obrade i donošenja novih zaključaka na osnovu tih podataka.

Ključni detalj u izgradnji sistema za analitiku jeste da je neophodno preskočiti jezero podataka kako bi smanjili vreme koje protekne od generisanja događaja do trenutka kad on postane dostupan za analizu i upotrebu. *Cloud Storage* iako sposoban za pouzdano i jeftino čuvanje velike količine podataka, nije najbolji izbor za skladištenje podataka ako želimo rezultate analize u realnom vremenu [17]. Prvi servis koji prihvata striming podatke u našoj platformi jeste *Cloud Pub/Sub,* koji kao što je napomenuto ranije, predstavlja dobar izbor za kratkotrajno čuvanje događaja. Nakon što događaji dospeju u platformu kroz *Pub/Sub*, ti podaci moraju da se obrade i za te potrebe može da se koristi ranije pomenuti servis *Cloud Dataflow,* koji potpomognut *Apache Beam* frejmvorkom omogućava obradu striming podataka u realnom vremenu [25]. *Dataflow* daje mogućnost agregiranja, filtriranja i grupisanja događaja pre nego što završe u skladištu kao što je *BigQuery*. Striming podrška *BigQuery* servisa za rezultat ima posledicu da obično za manje od minut nakon generisanja događaja oni budu obrađeni, uskladišteni i spremni za analizu [15]. Ukoliko želimo koristiti striming podatke za analizu vremenskih serija, podaci se iz *Dataflow* servisa mogu direktno učitati u *Cloud Bigtable*, skalabilnu, NoSQL bazu podataka namenjenu za izvršavanje analitičkih upita nad velikim podacima [24].

Iako se teži da se smanji vreme koje protekne dok se događaji ne pojave u skladištu podataka, dobra je praksa da se svi događaji u sirovom obliku iz *Dataflow* servisa takođe prosleđuju u jezero podataka. Na ovaj način organizacija će uvek imati rezervni izvor događaja koji mogu poslužiti u svrhe bekapa ili oporavka od greške, ali i izvođenja novih eksperimenata nad sirovim događajima kako bi se uvideo potencijal neke nove tehnike obrade događaja.



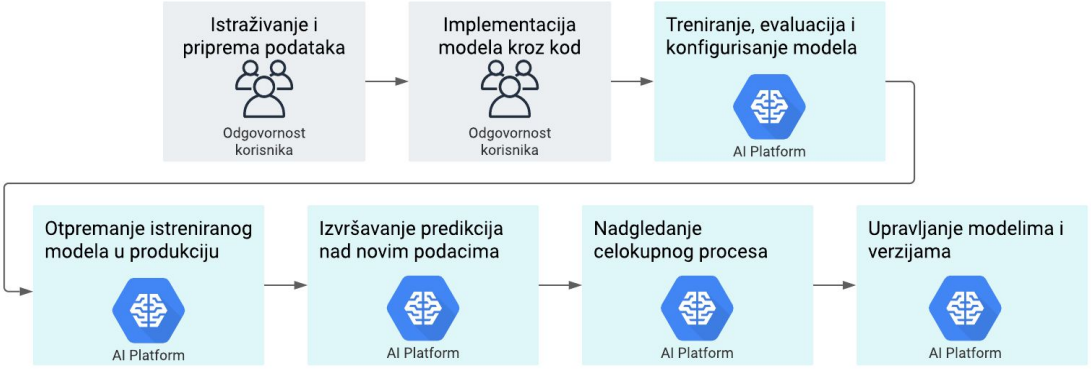
Slika 22. Proces analitike u realnom vremenu na *Google Cloud* platformi

4.6 Veštačka inteligencija i mašinsko učenje

Mašinsko učenje može da ima velike koristi od ogromne količine podataka uskladištenih u jezeru podataka. Tok rada jednog projekta mašinskog učenja zna da bude kompleksan i obuhvata veći broj koraka kao što su čišćenje i priprema podataka, kreiranje modela, treniranje modela i otpremanje modela u produkciju. Priprema sirovih podataka iz skladišta za upotrebu od strane nekog servisa za mašinsko učenje može se obaviti upotrebom istih alata i sistema kao i za beč analitiku, gde je ključni servis *Dataproc* koji omogućava obradu podataka preko alata iz *Hadoop* ekosistema. Ukoliko su modelu mašinskog učenja potrebni sveži striming podaci onda se može iskoristiti deo platforme koji se bavi analitikama u realnom vremenu i prosleđivati podaci iz *Dataflow* servisa koji se bavi obradom striming podataka.

Klaud provajderi danas ulažu velika sredstva u razvoj servisa koji će mašinsko učenje i veštačku inteligenciju učiniti dostupnim manjim i srednjim organizacijama. *Google Cloud* u svojoj ponudi ima veliki broj servisa za veštačku inteligenciju i neki od najpoznatijih su [22]:

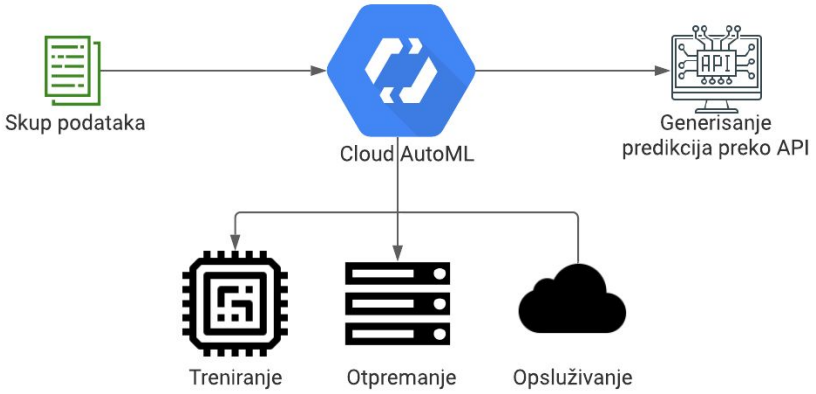
- ***AI Platform (Cloud Machine Learning)*** - Platforma za izradu, otpremanje i upravljanje modelima mašinskog učenja [36]. *AI Platform* zapravo je sačinjen od skupa servisa gde je svaki od njih zadužen za jedan korak u procesu razvijanja modela. *Google Cloud* je na sebe preuzeo upravljanje infrastrukturom i omogućio krajnjim korisnicima da se fokusiraju na izgradnju svojih modela. U procesu razvoja modela mašinskog učenja, korisnici su odgovorni za pripremu podataka i pisanje koda modela, a *AI Platform* je odgovoran za treniranje, evaluaciju i konfiguraciju modela [36]. Nakon što je model spreman, *AI Platform* omogućava jednostavno otpremanje i izvršavanje onlajn i beč predikcija, uz konstatno nadgledanje celokupnog procesa i verzionisanje modela. Ova platforma omogućava ekspertima mašinskog učenja da se fokusiraju na kreiranje svojih modela uz automatizaciju repetitivnih koraka i samoodrživu infrastrukturu.



Slika 23. Pregled rada *AI Platform* servisa

- ***Cloud AutoML*** - Predstavlja skup servisa koji omogućavaju inženjerima sa ograničenim iskustvom u mašinskom učenju da istreniraju visoko kvalitetne modele, prilagođene njihovom poslovnom problemu [35]. *AutoML* se oslanja ne *Google*-ovu industrijski vodeću tehnologiju "prenosa učenja" i pretrage neuralne arhitekture. Za razliku od *AI Platform* gde korisnici samo pišu kod za modele, kod *AutoML* korisnici obezbede skupove podataka za trening, a servis je zadužen za odabir pogodnog modela kroz napredne algoritme pretrage i prenosa učenja [35]. *Cloud AutoML* spaja prednosti automatizovanih alata za mašinsko učenje, gde nije potrebno pisati kod za sopstvene

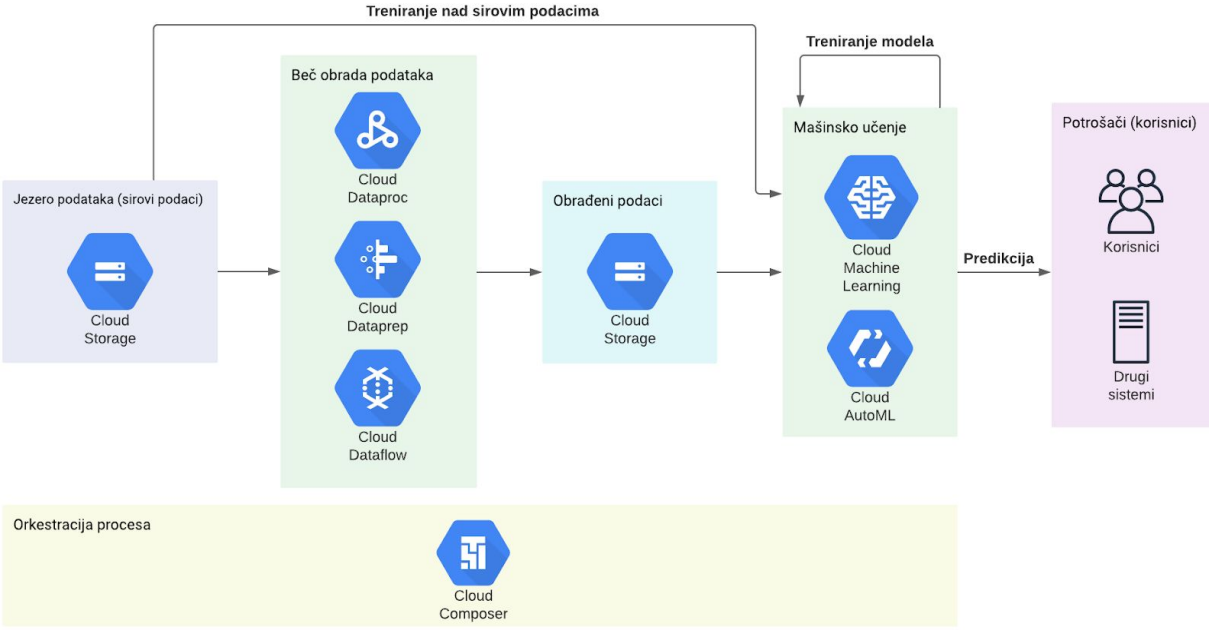
modele i sveta modela prilagodljivih specifičnom problemu, tako što omogućava korisniku da odabere skup podataka na kojem će se trenirati model. Na ovaj način se već postojeća ekspertiza u domenu mašinskog učenja primenjuje na specifičan problem koji ima neka organizacija.



Slika 24. Pregled rada *Cloud AutoML* servisa

- ***API-orjentisani AI servisi*** - Drugi naziv za ove servise jeste gradivni blokovi veštačke inteligencije u oblaku. Ovo je skup već istreniranih modela mašinskog učenja koji poseduju izuzetnu sposobnost rešavanja određenog problema [37]. Primer ovakvog servisa je *Cloud Vision* koji u suštini predstavlja već istreniran model mašinskog učenja predstavljen preko API koji prihavata sliku kao ulaz i kao izlaz vraća raznovrsne podatke kao što su prepoznati objekti na slici, određivanje pozicije objekata, prepoznavanje ljudi, prepoznavanje teksta, cenzurisanje neprikladnog sadržaja i slične informacije.

Momenat kada organizacija pređe sa jednostavnih deskriptivnih analitika koje analiziraju prošlost na sofisticirane tehnike prediktivne analitike koje prognoziraju budućnost predstavlja prekretnicu u razvoju i otvara vrata novim poslovnim prilikama i rapidnom rastu poslovanja [2].



Slika 25. Mašinsko učenje na *Google Cloud* platformi

5 Zaključak

Analitika podataka ima potencijal da promeni način na koji ljudi i organizacije pristupaju rešavanju problema i odlučivanju. Ukoliko govorimo o sportu, o poslovnom svetu ili o svakodnevnim aktivnostima u ljudskom životu, analitika podataka je promenila način na koji svet funkcioniše. Danas, osim što je glavni deo većine organizacija, analitika se koristi za razvijanje novih vrsta veštačke inteligencije, praćenje razvoja bolesti, razumevanje korisničkog ponašanja i pronalaženje slabih tačaka konkurencije u sportu i politici. Da bi izvukle maksimalne rezultate iz analitike, kompanije moraju da na pravi način primene nove tehnologije, da poboljšaju kvalitet svojih podataka i da efektivno upravljaju celim procesom obrade i analize. Organizacije koje budu u stanju da ispune ove preduslove, znatno će povećavati svoje šanse za uspeh i prednost nad konkurencijom, a sa druge strane, organizacije koje budu imale nepotpune ili minimalne platforme za analitiku, neće biti u mogućnosti da praktikuju donošenje odluka vođeno podacima i moraće se osloniti na lično iskustvo i osećanja pri određivanju pravca kojim će se kretati u budućnosti.

Tema ovog rada je izabrana na osnovu industrijskog iskustva koje posedujem u oblasti analitika, ali i zbog velikog ličnog interesovanja za moderne klaud tehnologije kao i tehnologije skladištenja, obrade i analize podataka. Revolucija računarstva u oblaku donosi nove servise koje je potrebno istražiti i adekvatno upotrebiti za rešavanje određenog problema. Pravilnom arhitekturom klaud sistema, obrada velike količine podataka i donošenje novih zaključaka i uviđanja na osnovu njih, postaju dostupni malim i srednjim organizacijama sa ekonomskog i tehnološkog aspekta. S obzirom na obim tehnologija, kao i brzi razvoj novih klaud servisa, oblast analitike podataka zahteva konstantno izučavanje i istraživanje.

Originalni doprinos ovog rada je sistematizacija dobrih praksi iz industrije, proučavanih kroz radove iz literature, i kombinovanje sa ličnim industrijskim iskustvom u svetu analitike kao i sa znanjem stečenim u toku studija, kako bi se predočili glavni problemi pri radu sa velikim podacima i predložio jedan pristup izrade arhitekture moderne platforme za analitiku, spremne da se izbori sa velikim problemima koje donose veliki podaci. Na početku rada se opisuju tradicionalna rešenja za analitiku podataka i razmatraju se njihove mane i razlozi koji su doveli do razvoja novih tehnologija i pristupa. U radu su detaljnije obrađeni servisi iz *Google Cloud* ponude, a potencijalni dalji koraci bi mogli da budu: detaljnije istraživanje i upoređivanje servisa za analitiku koje nude ostali veliki klaud provajderi, implementacija platforme za analitiku na osnovu definisane arhitekture na konkretnom primeru jedne organizacije.

Razvoj računarstva u oblaku sa sobom donosi i veliku priliku za sve vrste organizacija. Prilikom odabira servisa za implementaciju platforme za analitiku, potrebno je posebno obratiti pažnju na zahteve koje organizacija poseduje (npr. količina podataka, brzina podataka, format podataka ili izvori podataka), i na osnovu tih zahteva konstruisati platformu uz pomoć adekvatnih servisa. U suprotnom, ako se ne odabere pravi servis za pravu svrhu, može doći do izuzetno velikih finansijskih troškova za klaud servise. Iako klaud provajderi danas imaju veliki broj servisa u svojoj ponudi, koji na prvi pogled rešavaju sve probleme jedne organizacije, i dalje su organizacijama potrebne klaud arhitekte, sa velikim iskustvom i širokim znanjem u ovoj oblasti, kako bi zahteve jedne organizacije preneli na arhitekturu ekonomične platforme u oblaku. Veliki izbor raznovrsnih klaud servisa, sa sobom nosi i odgovornost odabira pravog alata za rešavanje

problema. Greška koju organizacije danas često ponavljaju jeste da svoju infrastrukturu iz centra podataka, u gotovo identičnom obliku, prenesu na okruženje u oblaku. Na ovaj način, organizacije gube neke od najvećih prednosti koje klaud danas nudi, kao što su: *serverless* (upotreba servisa bez održavanja servera), MPP (*Massive Parallel Processing*) klaud skladišta podataka, potpuno automatizovana elastičnost i skalabilnost, kao i mnoge druge prednosti.

U nastavku svoje karijere bih voleo da nastavim da se razvijam u oblasti analitike podataka, sa fokusom na moderne platforme u oblaku. Cilj mi je da nadogradim svoje znanje o računarstvu u oblaku, izučavajući i upotrebljavajući različite klaud servise kroz implementaciju sistema za prikupljanje, obradu, analizu i prezentovanje podataka. Nastojaću da steknem iskustvo u radu sa većim broj servisa iz *Google Cloud Platform* i *Amazon Web Services* ponude, kao i da potencijalno proširim svoje znanje o oblaku, učeći o ponudi i servisima ostalih provajdera.

6 Literatura

[1] D. Zburivsky, L. Partner, Designing Cloud Data Platforms, ISBN 9781617296444, Manning Publications, 2020.

[2] H. Nguyen, H. Pham, C. Chinm The Analytics Setup Guidebook, Holistics Software, 2020.

[3] V. Lakshmanan, Data Science on the Google Cloud Platform, ISBN-10 1491974567, O'Reilly Media, 2018.

[4] M. Kleppman, Designing Data-Intensive Applications, ISBN-10 1449373321, O'Reilly Media, 2017.

[5] Amazon Web Services. Big Data Analytics Options on AWS

[6] Google Cloud Platform. Smart Analytics on GCP

[7] Medium Blog. Building a Modern Data Analytics Platform in the Cloud

[8] New York Times The Age of Big Data

[9] Michigan State University. 4 Types of Data Analytics and How to Apply Them

[10] 10 Key Marketing Trends For 2017, IBM Cloud

[11] Statista. Internet of Things (IoT) connected devices installed base worldwide

[12] Big Sky Associates. The Data Analysis Process: 5 Steps To Better Decision Making

[13] The Age of Analytics: Competing in a Data-Drive World. 2016. McKinsey & Company

[14] Talend. Data Lake vs Data Warehous

[15] Google Cloud Platform. BigQuery

[16] Google Cloud Platform. Cloud Composer

[17] Google Cloud Platform. Cloud Storage as a data lake

[18] D. Vujošević, I. Kovačević, M. Suknović, N. Lalić. A comparison of the usability of performing ad hoc querying on dimensionally modeled data versus operationally modeled data. Elsevier, 2012.

[19] D. Vujošević, I. Kovačević, M. Vujošević-Janičić. The learnability of the dimensional view of data and what to do with it. Aslib Journal of Information Management, 2018.

[20] R. Kimball, M. Ross, The Data Warehouse Toolkit, ISBN-10 : 1118530802, Wiley, 2013.

[21] Google Cloud Platform. Cloud Data Fusion

[22] Google Cloud Platform. Cloud AI

[23] Apache. Apache Kafka Documentation

[24] Google Cloud Platform. Cloud Dataflow

[25] Google Cloud Platform. Cloud Pub/Sub

[26] Google Cloud Platform. Cloud Dataproc

[27] Google Cloud Platform. Cloud Dataprep

[28] Google Cloud Platform. Cloud Functions

[29] Google. Data Studio

[30] Google Cloud Platform. Cloud Functions

[31] Apache. Airflow Documentation

[32] Jupyter. Jupyter Documentation

[33] Google Cloud Platform. Cloud Datalab

[34] Towards Data Science. No-code Data Pipelines

[35] Google Cloud Platform. Cloud AutoML

[36] Google Cloud Platform. Cloud AI Platform

[37] Google Cloud Platform. Cloud Vision AI

7 Biografija

Branko Fulurija

Rođen 10. oktobra 1997. godine u Užicu. Završio srednju školu JUSŠC “Ivo Andrić” u Višegradu, smer opšta gimnazija. U toku srednje škole učestvovao na takmičenjima iz programiranja na svim nivoima. Postao višestruki regionalni, republički i državni prvak u programiranju u Bosni i Hercegovini i bio član tima koji je predstavljao BiH na Svetskoj informatičkoj olimpijadi. Osvajač bronzane medalje na Balkanskoj informatičkoj olimpijadi, održanoj na Kipru 2016. godine. Visoko obrazovanje započinje 2016. godine, kada kao stipendista upisuje Računarski fakultet u Beogradu, smer računarske nauke. Odradio dve studentske prakse kao softverski inženjer u “Majkrosoft” razvojnom centru u Beogradu, prvu 2017. godine, a drugu 2018. godine. Prvo zaposlenje kao softverski inženjer dobija u kompaniji “Kumulus Soft”, specijalizovanoj za pružanje konsultantskih usluga razvoja softvera u oblaku. Trenutno radi kao inženjer obrade i analize podataka u kompaniji “Nordeus”, nezavisnoj tehnološkoj kompaniji koja se bavi razvojem igara za mobilne telefone.