LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

LIELO DATU UN BLOKĶĒDES TĪKLA INTEGRĀCIJA

BAKALAURA DARBS

Autors: **Margarita Parhomenko**
Studenta apliecības Nr. : mp18051
Darba vadītājs: prof., Dr.sc.comp. Ģirts Karnītis

RĪGA 2022

ANOTĀCIJA

Mūsdienās lielie uzņēmumi un organizācijas plaši izmanto Lielo datu tehnoloģiju, lai uzkrātu, uzglabātu un analizētu datus. Tomēr Lielo datu tehnoloģija saskaras ar dažām problēmām, piemēram, datu vākšana, datu glabāšana, datu drošība, datu koplietošana. Savukārt pasaulē popularitāti gūst jauna decentralizētās datu uzglabāšanas tehnoloģija – blokķēde. Darba ietvaros tiks apskatīts un analizēts, kā blokķēdes un Lielo datu tehnoloģiju integrācija var uzlabot dažas Lielo datu problēmas.

Atslēgvārdi**:** blokķēde, Lielie dati, decentralizēta Lielo datu glabāšana, Lielo datu drošība, Lielo datu kvalitāte

ABSTRACT

BIG DATA AND BLOCKCHAIN NETWORK INTEGRATION

Today, large companies and organizations make extensive use of Big Data technology to collect, store, and analyze data. However, Big Data technology faces some challenges such as data collection, data storage, data security, data sharing. In turn, a new decentralized data storage technology – blockchain, is gaining popularity in the world. The paper will examine how the integration of blockchain and Big Data technologies can solve some Big Data problems.

Keywords: blockchain, Big Data, decentralized Big Data storage, Big Data security, Big Data quality

SATURS

APZĪMĒJUMU SARAKSTS

Bitcoin - mūsdienās populārākā un lielākā decentralizētā digitālā valūta (kriptovalūta).

Ethereum - decentralizēta atvērtā koda blokķēde ar viedā līguma funkcionalitāti.

IDC - International Data Corporation.

ZB (zettabyte) – zetabaits. 1 ZB = 1021 baiti.

PB (petabyte) – petabaits. 1 PB = 1015 baiti.

EB (exabyte) – eksabaits. 1 EB = 1018 baiti.

GB (gigabyte) – gigabaits. 1 GB = 109 baiti.

TB (terabyte) – terabaits. 1 TB = 1012 baiti.

USD - United States dollar

Statista - Vācijas uzņēmums, kas specializējas tirgus un patērētāju datos.

CAGR (compound annual growth rate) – salikts gada pieauguma rādītājs, parāda, par cik procentiem pieaug pētītais parametrs gadā.

Mining – rakšana. Blokķēdē tas ir process, kura rezultātā tiek ražotas jaunas monētas un validētas jaunas transakcijas.

Maineris (angliski miner) – blokķēdes tīkla dalībnieks (mezgls), kurš konkurē ar citem maineriem par tiesībām izveidot jaunu bloku.

Mezgls (angliski node) – ierīce (piemēram, dators), kas ir blokķēdes tīkla sastāvdaļa.

Full node – dators, kas ir savienots ar blokķēdes tīklu un pārbauda visus blokķēdes noteikumus un satur visu blokķēdes virsgrāmatas kopiju.

Mempool - saglabā informāciju par neapstiprinātām transakcijām, kas vēl nav iekļauti blokā.

ISO (International Organization for Standardization) - ir starptautiska standartu izstrādes organizācija, kas sastāv no dalībvalstu nacionālo standartizācijas organizāciju pārstāvjiem.

MITRE – Amerikāņu organizācija, kas pārvalda federāli finansētus pētniecības un attīstības centrus.

ASIC (application-specific integrated circuit) - ir integrētās shēmas mikroshēma, kas ir izstrādāta, lai pēc iespējas ātrāk atrisinātu neuzlabotā PoW algoritma aprēķinus. Izmanto bitkoinu rakšanai.

IoT (Internet of Things) - ierīču grupa ar sensoriem, kas ir apvienoti vienā tīklā un var vākt dažādus datus no reālās pasaules.

TPS (transactions per second) – transakciju skaits, ko tīkls spēj apstrādāt katru sekundi.

DON (decentralized oracle network) - decentralizēts orākula tīkls.

DLD – decentralizēti Lielie dati.

TS (Threshold Signature) - sliekšņa paraksts.

P2P (peer-to-peer) network – vienādranga tīkls, tīkls kur visi mezgli ir savstarpēji saistītas un tām ir vienādas tiesības.

CPU (central processing unit) – centrālais procesors.

CRUD (create-read-update-delete) - galvenās operācijas, ko izmanto datu bāzes, API vai lietotāju saskarnes.

SQL (Structured Query Language) - strukturētā vaicājumu valoda.

NoSQL (Non-Structured Query Language) – sistēma, kas ļauj pārvaldīt nerelāciju datus.

XML (Extensible Markup Language) – iezīmēšanas (angliski markup) valoda, ko izmanto datu serializācijai.

JSON (JavaScript Object Notation) – atvērtā standarta faila formāts, ko izmanto datu serializācijai.

DES (Data Encryption Standard), AES (Advanced Encryption Standard) - simetriskas atslēgas algoritmi.

SHA-3 (Secure Hash Algorithm 3), SHA-256 (Secure Hash Algorithm 256) – jaucējfunkcijas.

IEVADS

Lielo datu apjoms katru gadu strauji pieaug. Vācijas datu analīzes uzņēmuma Statista 2022. gadā publicētajā pētījumā teikts, ka 2020. gadā publicētais digitālo datu apjoms varētu būt 64.2 ZB, savukārt 2025. gadā prognozētais datu apjoms ir 181 ZB [5]. Šādi dati var spēcīgi ietekmēt daudzas nozares. Jo vairāk informācijas mums ir, jo precīzāk varam paredzēt nākotni vai izdarīt svarīgus secinājumus par pagātni.

Palielinoties lielo datu apjomam, uzņēmumiem kļūst grūtāk uzraudzīt šādu datu kvalitāti un uzturēt kvalitātes līmeni tajā pašā līmenī. Tāpat, pieaugot apjomam, rodas problēmas arī ar datu privātuma saglabāšanu. Dati nepārtraukti migrē, tiek apvienoti, un šādos apstākļos kļūst grūti nodrošināt augstu drošību un privātumu. Problēmas ir arī ar datu kvalitāti, iegūtie dati bieži vien ir nepilnīgi un neprecīzi un pretrunīgi.

Pirmā un otrā blokķēdes paaudzes neļāva strādāt ar tik lielu datu apjomu – tām ir pārāk zema mērogojamība un transakciju ātrums. Bet līdz ar trešās paaudzes blokķēdes parādīšanos šāda iespēja ir parādījusies. Blokķēde 3.0 izmanto gan viedos līgumus, gan konsensa mehānismus, tie var novest pie risinājuma, kas ir pietiekami mērogojams, ar lielu transakciju ātrumu, drošs un atvieglo manipulāciju ar Lieliem datiem.

Šī darba mērķis ir izpētīt iespējas risināt aktuālas lielo datu problēmas, izmantojot blokķēdes tehnoloģiju, noteikt decentralizēto lielo datu stiprās un vājās puses.

Šī darba ietvaros tika arī izmantots šī autores iepriekš izstrādātais kursa darbs [15], kurā rezultātā tika analizēta blokķēdes ieviešanas iespēja piegādes ķēdēs, minētā darba gaitā arī tika analizēta blokķēdes tehnoloģija un lielāko daļu no blokķēdes definīcijas var atrast tajā darbā.

Šī darba ietvaros tika apskatīti zinātniskie darbi un raksti, analizēta mūsdienās aktuālā statistika un analizēta dažādu decentralizēto krātuves projektu dokumentācija.

Darba laikā tiek definēti galvenie parametri, kas nepieciešami, lai izveidotu decentralizētas Lielo datu krātuves. Tiek apsvērtas dažas tehnoloģijas un arhitektūras risinājumi, kas ļauj uzglabāt datus lielā apjomā blokķēdē. Tiek noteiktas problēmas, ko blokķēdes tehnoloģija var atrisināt lielo datu jomā.

Darba pirmajā nodaļā apskatīta Lielo datu jēdziens, problēmas un centralizētie risinājumi Lielo datu apstrādei un uzglabāšanai. Otrajā nodaļā ir apskatīta Blokķēdes jēdziens, galvenās iezīmes un blokķēdes spēja apstrādāt Lielos datus. Trešajā nodaļā apskatīti esošie decentralizēti Lielo datu apstrādes risinājumi, to darbības princips. Tiek apskatīts arī orākulu darbības princips un blokķēdes spēja atrisināt Lielo datu problēmas.

1. LIELO DATU JĒDZIENS UN PROBLĒMAS

Lielie dati tiek uzskatīti par datu kopām, kurām ir ļoti liels izmērs un kuras nevar uzglabāt, pārvaldīt un analizēt ar tradicionālajiem datu bāzes rīkiem [6]. Saskaņā ar [7], Lielo datu nosaka 3 (saskaņā ar citiem avotiem – 4 [11]) galvenās pazīmes:

- Tilpums (angliski *volume*) – datu daudzums, piemēram, tiešsaistes kameras pilsētā katru dienu ražo milzīgu datu apjomu.
- Ātrums (angliski *velocity*) - ātrums, ar kādu tiek saņemti dati. Kameras sniedz informāciju nemainīgā ātrumā.
- Šķirne (angliski *variety*) - dažādi datu avoti (attēli, video, audio, sensoru dati un tā tālāk). Mūsdienās ir jāpaļaujas uz dažādiem datu avotiem, bieži dati tiek saņemti nestrukturēti un netiek saņemti saskaņotā veidā.
- Vērtība (angliski *value*) - dati paši par sevi, neatkarīgi no to apjoma, parasti nav īpaši noderīgi, tie ir jāapstrādā un jāanalizē, lai iegūtu vērtīgu informāciju.

Lielie dati ir dati, kas sastāv no dažādiem datu tipiem un tiek ņemti no dažādiem datu avotiem lielā apjomā ar lielu ātrumu.

Lielie dati ir pastāv gandrīz visās mūsu dzīves jomās, to izplatīšanu izraisa daudzi faktori: Interneta lietotāju skaits, informācija uz vienu iedzīvotāju visā pasaulē, Internetam pieslēgto ierīču skaits, datu centru skaits, un tā tālāk [8]. Zemāk ir minēti daži Lielo datu izmantošanas piemēri:

- Sociālie tīkli. Facebook serveri vāc attēlu, video, teksta datus. Twitter vāc datus par īsziņām, un katru minūti YouTube tiek pievienots 72 stundu video (2017) [7].
- Banku darījumi. Bankas vāc lielus datu apjomus, kas jāapstrādā droši un tiem jābūt uzticamiem, jo balstoties uz tiem datiem var konstatēt krāpšanas gadījumus, piemēram, kad tiek izmantota zagta kredītkartes informācija. Lai konstatētu tādu krāpšanu, būs jāanalizē liels datu apjoms, kur katra transakcija tiek saņemta ātri, un lēmums ir jāpieņem, tiklīdz tiek saņemta transakcija.
- Piegādes ķēdes. Interneta veikals Amazon miljoniem klientu īsā laika periodā veic ļoti lielu transakciju skaitu. Amazon jāanalizē dati, lai sniegtu lietotājiem labāko produktu piedāvājumu vai optimālāko reklāmu.

- Internets. Tīmekļa vietnes glabā lielo datu apjomu, lai pārlūkprogramma varētu analizēt un atgriezt pieprasījumam atbilstošo rezultātu, tīmekļa pārmeklētājs indeksē lapas un vietnes dati tiek saglabāti datubāzē kā modeļi.
- Veselības aprūpe. Slimnīcas glabā pacientu datus, Lietu internets (IoT) (viedās ierīces) sniedz cilvēku veselības datus.
- IoT (angliski *Internet of Things*) - dažādi sensori, piemēram, medicīnas ierīces, apkopo informāciju un sūta datus sistēmām, kas palīdz analizēt datus. IoT ierīču apkopotos datus var uzskatīt par lielu datu kopu.

1.1. Lielo datu tipi

Dati var būt pilnīgi dažāda veida, piemēram, fotoattēli, e-pasta ziņojumi, faili, tabulas, metrika no IoT ierīces, vai arī lietojumprogrammu dati (CRUD darbības). Parasti datus iedala 3 veidos:

- Strukturēti dati

Strukturēto datu piemērs ir skaitļi, virknes, datumi, utt. Par strukturētiem datiem uzskata tos datus, kuriem ir definēta noteikta shēma un atribūtu kopums. Strukturētos datus sauc arī par relāciju datiem. Lai apstrādātu strukturētos datus, ir nepieciešama strukturētā vaicājumu valoda (SQL). Šos datus ir viegli analizēt.

- Daļēji strukturēti dati

Daļēji strukturētiem datiem nav definēta stingri noteikta apstrādes vai uzglabāšanas shēma. Tie nav relāciju dati, tiem nav struktūras, tomēr daļēji strukturēti dati parasti satur pazīmes, kas palīdz atšķirt dažādas entītijas vienu no otras, piemēram, atslēgu-vērtību (angliski *key-value*) pāri. Daļēji strukturēto datu apstrādei nevar izmantojot SQL valodu. Datu apstrādei, glabāšanai, sūtīšanai izmanto datu serializācijas valodas, piemēram, XML, JSON, utt.

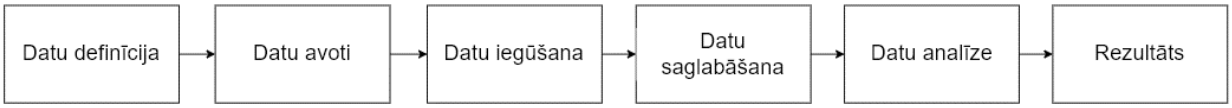
- Nestrukturēti dati

Nestrukturēto datu piemērs ir fotoattēli, video, teksta dokumenti, utt. Tādiem datiem nav noteiktas shēmas. Nestrukturēti dati nevar tikt glabāti parastajos relācijas datubāzēs un tikt apstrādāti ar SQL valodu.

1.2. Lielo datu dzīves cikls

Lielo datu dzīves cikls ir sarežģīts process, kas parasti sākas ar biznesa problēmas identificēšanu, vajadzīgās informācijas noteikšanu, datu vākšanu un beidzot ar rezultāta analīzi un publicēšanu. Tālāk ir apskatītas Lielo datu dzīves cikla galvenie posmi (sk. 1.1. attēlu).

- Datu definīcija – tiek definētas biznesa problēmas, tiek noteikti datu avoti un datu kvalitātes prasības.
- Datu avoti - mākoņkrātuves, IoT ierīču dati, utt. Dati var būt strukturēti, daļēji strukturēti un nestrukturēti.
- Datu iegūšana - Lielie dati pārsvarā ir nestrukturēti, šajā posmā dati tiek filtrēti, piemēram, tiek noņemti bojāti vai neatbilstoši dati, kas neatbilst analīzes mērķim, pēc tām dati var tiek notīrīti un validēti (piemēram tiek izmestas rindas, kuriem ir tikai NULL vērtības).
- Datu saglabāšana – dati tiek saglabāti. Pirms saglabāšanas, ja datu masīvs satur vairākas datu kopas, tad dažas no šīm datu kopām var tikt apvienotas, piemēram, izmantojot kopīgus laukus.
- Datu analīze – dati tiek pakļauti dažādām manipulācijām, piemēram, datizracei (angliski *data mining*), mašīnmācīšanas algoritmiem, lai atrastu konsekventus modeļus un šablonus.
- Rezultāts – rezultāti tiek apstrādāti, attēloti cilvēkam saprotamā veidā un izmantoti visādu procesu optimizācijai.



1.1. att: **Lielo datu dzīves cikls**

1.3. Datu kvalitāte

Aktīvus pētījumus par datu kvalitāti IT nozares profesionāli sāka veikt kopš 90. gadu sākuma. Vienu no šādiem pētījumiem veica MIT Universitātes grupa profesora R.Vanga vadībā

1996. gadā. Viņi definēja "datu kvalitātes dimensiju" jēdzienu – atribūts, kas atspoguļo vienu datu kvalitātes aspektu [12]. Tomēr Lielo datu kvalitātes dimensiju definīcijas atšķiras dažādos avotos. Zemāk ir minēti visbiežāk sastopamas datu kvalitātes dimensijas [10] [11] :

- Pilnīgums (angliski *completeness*) - vai komponenta trūkums ietekmēs datu precizitāti un integritāti. Datiem pēc iespējas ir jābūt pilnīgiem, ja iespējams, pilnībā jānorāda datu kopas metadati, piemēram, datums un atribūtu apraksts.
- Nepretrunīgums (angliski *consistency*) - datu savstarpēja konsekvence, datu integritāte.
- Precizitāte (angliski *accuracy*) - dati neizraisa neskaidrības, sniegtie dati ir precīzi un atbilst avotam.
- Piekļustamība (angliski *accessibility*) – ir nodrošināts interfeiss, caur kuru var viegli piekļūt datiem.
- Savlaicīgums (angliski *timeliness*) – laika intervāls no datu vākšanas un apstrādes līdz publicēšanai atbilst prasībām, dati tiek regulāri atjaunināti.

Taču šī ir tikai viena šādu kvalitātes dimensiju interpretācija, pētījumā [27] minēts, ka dažādos avotos dimensiju nozīme var tikt definēta dažādi, un tie var būt pretrunā viens otram. Katras dimensijas precīza nozīme joprojām tiek apspriesta un nav vienošanās par to nozīmi.

Tomēr konkrēta raksturlieluma ietekme uz datu kvalitāti ir atkarīga no konkrētā gadījuma. Augstākās kvalitātes dati būs tie, kas atbilst attiecīgajam lietojumam un atbilst prasībām, t.i. datu kvalitātes vērtējums ir atkarīgs no datu patērētājiem. Saskaņā ar ISO 9000 standartu un [27], datu kvalitāte ir atkarīga no tā, kā patērētājs definē kvalitāti un cik labi iegūtie dati atbilst šai definīcijai.

1.4. Lielo datu glabāšana

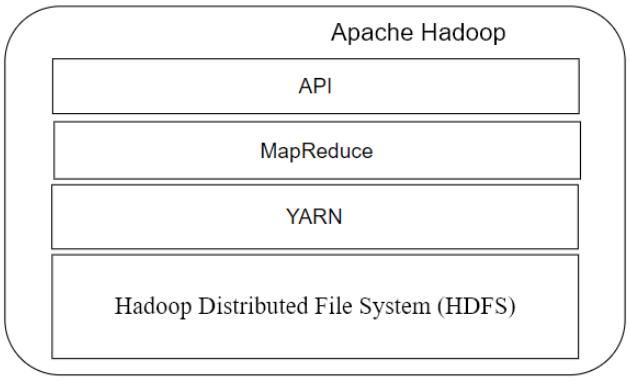
Tradicionālās relāciju datu bāzes, piemēram, Oracle, MySQL vai SQL, nevar apstrādāt tik milzīgu Lielo datu apjomu, kā arī nevar apstrādāt nestrukturētu datu formātu. Lielie dati tiek glabāti nerelāciju datubāzēs (NoSQL). Nerelāciju datubāzēs netiek izmantotas parastas tabulas, tā vietā NoSQL datu bāzes optimizē krātuves modeļus atbilstoši datu tipam.

Lielo datu glabāšanai ir daudz risinājumu, šādi risinājumi ietver mērogošanas iespēju, jo Lielo datu apjoms nepārtraukti pieaug, kā arī paredz iespēju nestrukturētu datu uzglabāšanai. Šajā nodaļā tiks aplūkots, kā Apache Hadoop uzglabā un apstrādā Lielos datus.

1.4.1. Apache Hadoop

Apache Hadoop ir atvērtā pirmkoda programmatūras utilītu kolekcija, kas ļauj glabāt un apstrādāt Lielos datus. Lielo datu apstrāde kļūst iespējama pateicoties klasteriem un paralēlai skaitļošanai.

Apache Hadoop sastāv no vairākiem moduļiem, izplatītās failu sistēmas, kurā tiek glabāti dati, YARN, kas pārvalda resursiem un MapReduce ietvaru, kas apstrādā datus (sk. 1.2. attēlu).



1.2. att: **Hadoop arhitektūra**

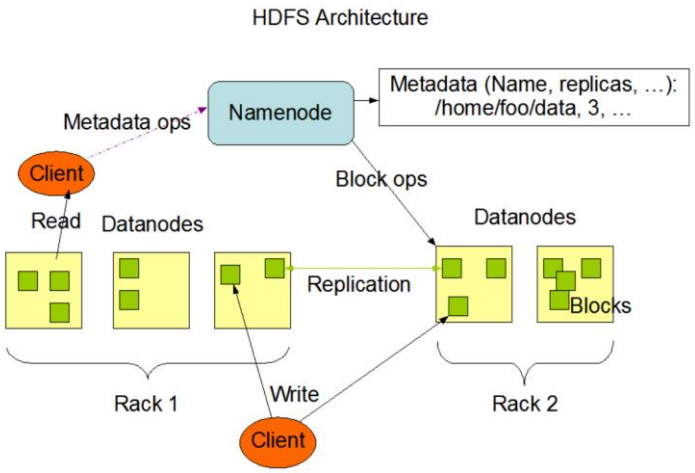
Hadoop ir *master-slave* tipa arhitektūra, kur par *master* mezgli kalpo *NameNode* un par *slave* mezgli – *DataNode* (sk. 1.3. attēlu). Katram Hadoop *DataNode* mezglam ir sava diska vieta, atmiņa, caurlaidspēja un apstrāde, t.i., savs dators.

Hadoop Distributed File System (HDFS)

HDFS ir izplatīta failu sistēma, kas ļauj glabāt lietotāja datus failos. Katrs fails tiek sadalīts vienā vai vairākos blokos un šie bloki tiek saglabāti sekundārās nodēs (DataNodes).

Galvenais centralizēts mezgls NameNode pārvalda failu sistēmu kopumu un regulē klientu piekļuvi failiem, arī tajos tiek glabāti visi HDFS metadati (failu nosaukumi, informācija par faila blokiem, bloku atrašanās vietas, atļaujas utt).

Sekundārie mezgli DataNodes, pārvalda vienu vai vairākas krātuves un izpilda galvenā NameNode mezgla norādījumus, piemēram, veic pieprasījumus [40].



1.3. att: **HDFS arhitektūra [40]**

Hadoop YARN (Yet Another Resource Negotiator)

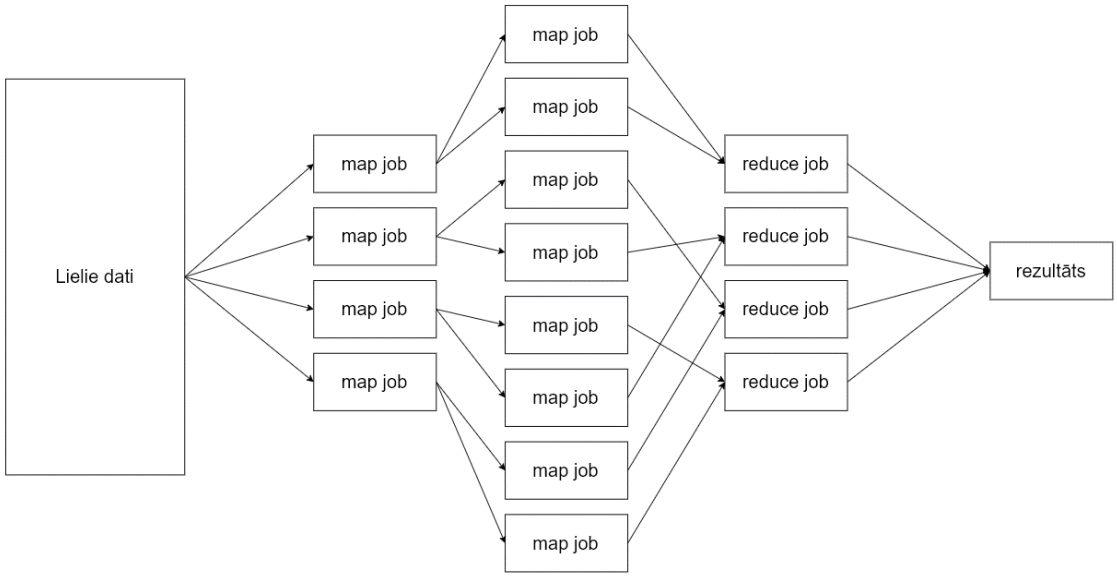
YARN nodrošina Hadoop resursu pārvaldību. Tam ir *master-slave* tipa arhitektūra, kas sastāv no resursu pārvaldnieku (master) (*Resource Manager* RM) un vairākiem mezgla pārvaldniekiem (slave) (*Node Manager* NM). RM atrodas tajā pašā mezglā, kur atrodas HDFS *NameNode*, attiecīgi, NM atrodas tajā pašā mezglā, kur atrodas *DataNode*.

RM uzdevums ir sadalīt resursus starp visām sistēmas lietotnēm, šis mezgls pārvalda NM. Savukārt, NM ir atbildīgs par savu mezglu, seko to resursu lietojumam (CPU, atmiņu, disku, tīklu) un ziņo par to RM [42].

Hadoop MapReduce

MapReduce ir programmatūras ietvars lietotņu rakstīšanai, kas apstrādā milzīgus datu apjomus (vairāku terabaitu datu kopas) paralēli vairākos mezglos (tūkstošiem) uzticamā un kļūdu izturīgā veidā [41].

Sākumā MapReduce izpilda *map job*, kas ņem datu kopu un pārvērš to citā datu kopā, kur atsevišķi elementi tiek sadalīti <*key*, *value*> pāros (sk. 1.4. attēlu). Pēc tām *reduce job* paņem *map job* rezultātu un apvieno šos datu <*key*, *value*> pārus mazākā <*key*, *value*> pāru kopā. Visas datu kopas tiek apstrādātas paralēli, izmantojot atsevišķu mezglu. Ja šī paradigma tiek implementēta blokķēdē, tad parasti jebkurš mezgls var paņemt datu kopu un sākt veikt aprēķinus.



1.4. att: **MapReducer darbības princips**

Hadoop ir *master-slave* tipa arhitektūra un tā trūkums ir centralizēta datu apstrāde (*NameNode* ir centralizēta).

1.5. Lielo datu izaicinājumi

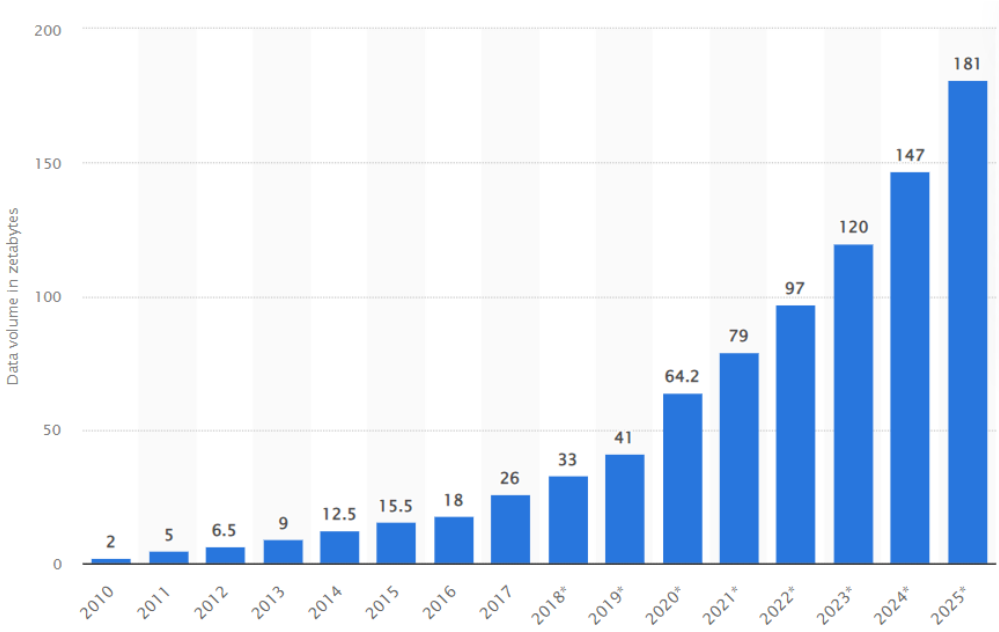
1.5.1. Datu kvalitāte

Datu avotu daudzveidība piegādā dažāda veida datus: attēlus, video, audio, sensoru datus un tā tālāk. Mums ir jāpaļaujas uz dažādiem datu avotiem. Bieži dati, kas tiek sniegti, ir nepilnīgi, trokšņaini un nestrukturēti.

Saskaņā ar 2016. gada statistikas aptauju [46], vispopulārākie sliktās datu kvalitātes iemesli Ziemeļamerikā ir (1) darbinieku nepareiza datu ievade, (2) datu migrācijas vai datu konvertācijas projekti, (3) jaukti ieraksti, ko veikuši vairāki lietotāji, (4) izmaiņas datu avota sistēmās, (5) sistēmu kļūdas. Tikai ļoti neliela daļa aptaujāto norādīja, ka ar datiem nav nekādu problēmu. Slikta datu kvalitāte ietekmē analīzes rezultātu, kā arī prasa papildu izmaksas.

1.5.2. Datu apjoms

Pieaugot datu apjomam, uzņēmumiem ir grūti nodrošināt datu kvalitāti. 2018.gadā IDC prognozēja ka ģenerēto digitālo datu apjoms pieaugs no 33 ZB (2018 gadā) līdz 175 ZB 2025.gadam un līdz 2025. gadam 49% no pasaulē saglabātajiem datiem atradīsies publiskā mākoņa vidē. [4]. Jaunākā Statista pētījumā (2022) [5], kura pamatā ir 2018. gadā publicētais ICD pētījums [4], tika analizēti dati no 2010. līdz 2020. gadam un prognozēts, ka nākamajos piecos gados, līdz 2025. gadam, globālā datu ģenerēšana pieaugs līdz vairāk nekā 180 ZB, kā parādīts 1.5. attēlā. Pieaugums bija lielāks nekā iepriekš gaidīts, tas ir saistīts ar COVID-19 pandēmijas sekām, jo vairāk cilvēku strādāja un mācījās no mājām un biežāk izmantoja mājas izklaides iespējas [5].

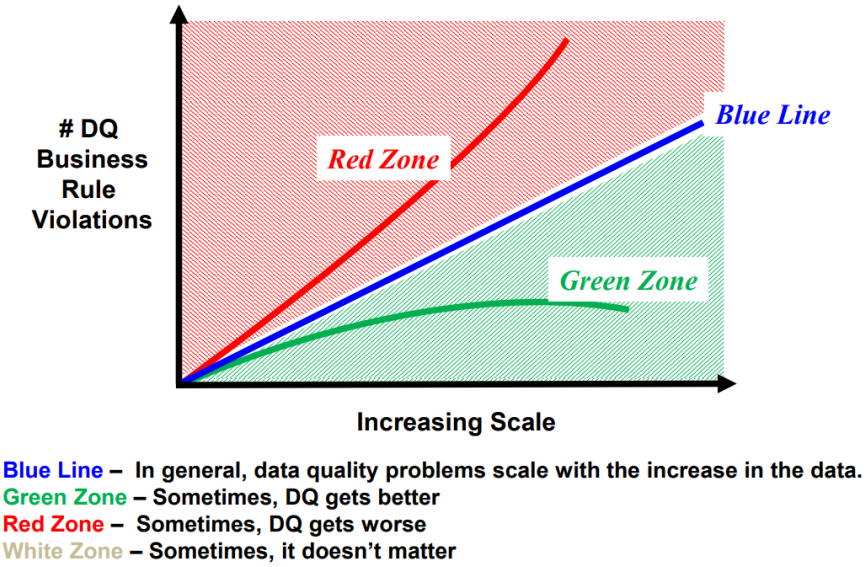


1.5. att: **Statista: ģenerēto digitālo datu apjoms no 2010. līdz 2020. gadam un prognozētais datu apjoms 2021. - 2025. gadam [5]**

Tomēr ne visi ģenerētie dati tiek saglabāti un ne visi dati tiek glabāti ilgu laiku. Šī pati publikācija apstiprina, ka tikai 2% no 2020. gadā saražotajiem un izmantotajiem datiem ir saglabāti un glabāti līdz 2021. gadam, kas ir aptuveni 1.3 ZB. Līdz ar prognozēto datu apjomu pieaugumu tiek arī prognozēts, ka laika posmā no 2020. līdz 2025. gadam datu krātuves ietilpība palielināsies, pieaugot ar CAGR 19.2% apmērā prognozētajā periodā no 2020. līdz 2025. gadam. 2020. gadā uzstādītā datu krātuves ietilpība sasniedza 6,7 ZB. [5]

Tā kā kopš 2020. gada ģenerēto datu apjoms ir strauji pieaudzis, var sagaidīt, ka tuvāko 5 gadu laikā pieaugs arī digitālās krātuves apjoms, lai gan digitālās krātuves apjoma pieaugums ir lēnāks nekā datu ģenerēšanas pieaugums. Ir svarīgi saglabāt pēc iespējas vairāk datu, jo saglabāto datu apjoms tieši ietekmē veikto pētījumu kvalitāti un kvantitāti.

2015. gada MITRE pētījumā [13] teikts, ka vairumā gadījumu palielinoties datu apjomam, pasliktinās datu kvalitāte (sk. 1.6. attēlu). Tas lielā mērā ir saistīts ar uzņēmumu nespēju kontrolēt arvien pieaugošā datu apjoma kvalitāti.



1.6. attēls: **Sakarība starp lielo datu apjomu un to kvalitāti [13]**

Tā kā Lielo datu apjoms ir milzīgs, ir grūti iegūt nepieciešamos augstas kvalitātes datus, kā arī novērtēt šādu datu kvalitāti saprātīgā laika posmā. Arī pastāv problēma ar Lielo datu apstrādi, jo nestrukturēto datu īpatsvars lielajos datos ir ļoti augsts, būs nepieciešams daudz laika, lai pārveidotu nestrukturētos tipus strukturētos tipos un tālāk apstrādātu datus [11].

1.5.3. Datu drošība un privātums

Datu drošība un privātums ir viens no svarīgākiem jautājumiem, jo Lielie dati bieži satur konfidenciālo informāciju, piemēram, pacientu veselības datos bieži minēts cilvēka vecums, personas kods, pases dati, adrese, utt.

Tradicionālie drošības risinājumi bija izgudroti un paredzēti neliela datu apjoma aizsardzībai un nav piemēroti Lielo datu apjomiem, piemēram, Hadoop izmanto vairākus mezglus (tūkstoši) datu glabāšanai, ir nepieciešams nodrošināt katra mezgla drošību.

DDoS uzbrukums

Viens no draudiem Lielajiem datiem ir DDoS (angliski *distributed denial-of-service*) uzbrukums. DDoS uzbrukumā uzbrucēji izmanto vairāku datoru apvienojumu, lai uzbruktu sistēmai. DDoS uzbrukumi noved pie situācijas, kad cietušais tīkls nespēj savlaicīgi saņemt un apstrādāt pieprasījumus, tādējādi traucējot normālu pakalpojumu sniegšanu lietotājiem [25]. Šāda uzbrukuma rezultāts var radīt sekojošas sekas:

- Vietne ir pārslogota un kļūst nepieejama
- Sistēma ir vairāk neaizsargāta pret uzbrukumiem, datu zuduma risks ir lielāks
- Var būt nepieciešams papildu laiks un nauda, lai atjaunotu normālu sistēmas darbību
- Datu vākšana tiek pārtraukta

Datu konfidencialitāte

- **Lietotāja dati viņam nepieder.** Dati, kas tiek glabāti centralizēti, pieder konkrētam uzņēmumam. Facebook vāc informāciju par to, kā mēs izmantojam savus pakalpojumus, piemēram, par satura veidiem, ko skatāmies vai ar kuru mijiedarbojamies, vai par mūsu darbību biežumu un ilgumu [29]**.** Tādējādi lietotājiem ir datu saglabāšanas tiesības (lietotājs var dod organizācijai tiesības uzglabāt viņa datus), nevis datu īpašumtiesības (dati nepieder lietotajam).
- **Melnās kastes problēma.** Mēs nezinām, kas īsti notiek ar mūsu datiem. Runājot par medicīniskiem ierakstiem, persona, par kuru tika veikts ieraksts, nevar būt pilnībā pārliecināta, ka organizācija nav kopīgojusi viņa datus ar kādu citu organizāciju/cilvēku, kā arī nevar to pilnībā kontrolēt. Facebook gadījumā, ja lietotājs piekrīt sīkdatņu lietošanai, šie dati var tiks izmantoti lai, piemēram, rādītu mums atbilstošas reklāmas vai saturu [30]. Problēma paliek tāda, ka mums ir jāuzticas uzņēmumam Meta, ka viņi mūsu datus tiešām nevienam nedod un neizmanto citiem mērķiem.
- **Daļēja piekļuve datiem.** Drošības nolūkos uzņēmumi un organizācijas ierobežo piekļuvi privātiem datiem. Bet šādos gadījumos dati parasti tiek pilnībā šifrēti, piemēram, medicīnas ieraksti, kas ietver personas informāciju. Šajā gadījumā var iedomāties situāciju, kad medicīnas pētnieks vēlas izmantot datus, kas nav saistīti ar personas privātumu, bet viņam šādas iespējas nebūs.

2. BLOKĶĒDES TEHNOLOĢIJAS JĒDZIENS

Blokķēdes galvenās sastāvdaļas un darbības princips tika analizēti un aprakstīti autora kursa darbā [15], tāpēc šajā nodaļā īsumā tiks aprakstītas blokķēdes galvenās sastāvdaļas un papildināti punkti, kas netika apskatīti minētajā darbā.

Mūsdienās blokķēdi var izmantot, lai uzglabātu un pārvaldītu jebkāda veida aktīvus vai datus. Aktīvus var iedalīt fiziskajos (angliski hard assets) un nemateriālajos (angliski intangible assets). Fiziskie aktīvi ir tas, ko var redzēt un kam var pieskarties – fiziskas lietas, piemēram, iekārtas, ēkas, transportlīdzekļi. Piegādes ķēdes ir viens no lietošanas piemēriem fizisko aktīvu glabāšanai blokķēdē, jo tie glabā datus par precēm, izejvielām, u.c. [15]. Nemateriālie aktīvi ir tie, kas fiziski neeksistē – piemēram, veselības dati, balsis, utt. Tādējādi blokķēdi var izmantot, lai savāktu, novērtētu un pārsūtītu ierobežotu daudzumu jebkuras informācijas.

2.1. Blokķēdes struktūra

Blokķēdei ir vairākas īpašības, kas to atšķir no citiem tehniskajiem risinājumiem (sk. 2.1. tabulu).

Īpašība	**Apraksts**
Nemainība	Blokķēdei pievienotos datus nevar mainīt vai dzēst.
Decentralizācija	Nav centrālās pārvaldes iestādes, kas būtu atbildīga par visiem lēmumiem.
Izplatītā virsgrāmata	Katram pilnajam tīkla mezglam ir sava virsgrāmatas kopija.
Drošība	Dažādu tehnoloģiju kombinācija neļauj viegli pievienot, mainīt vai noņemt datus no ķēdes.

2.1. tabula: **Galvenās blokķēdes īpašības**

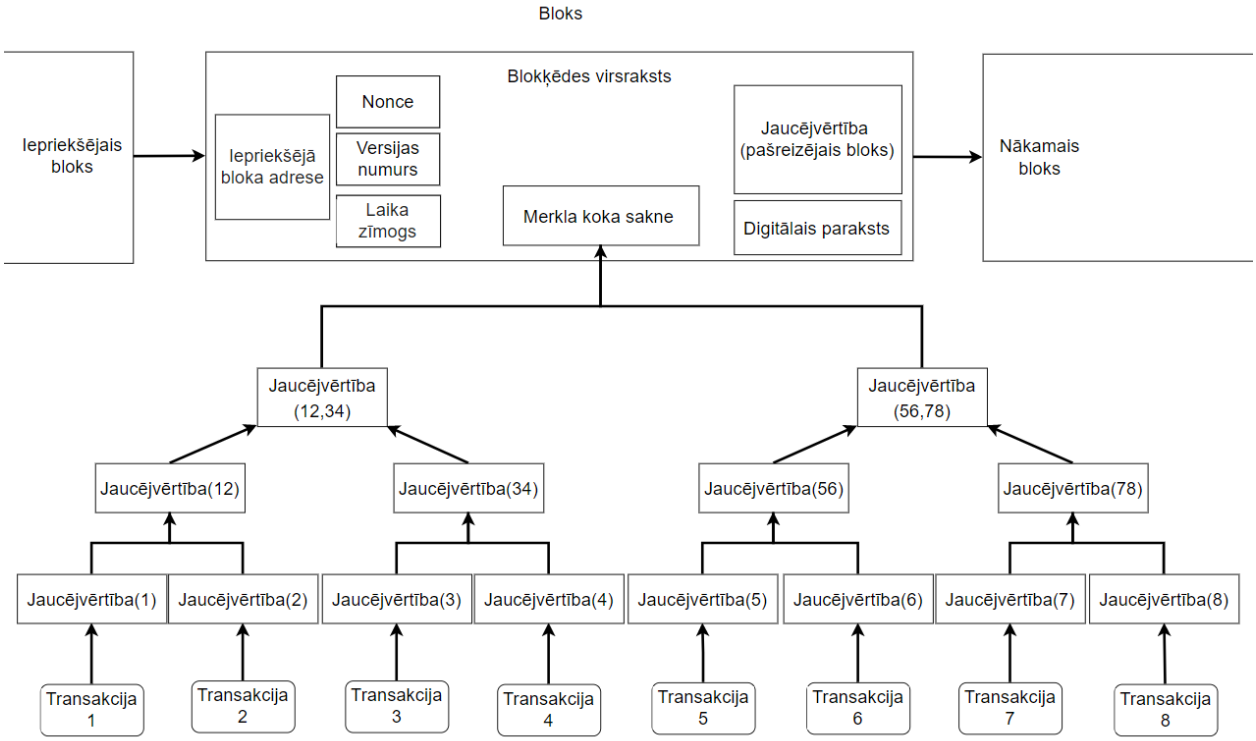
Šādas blokķēdes īpašības nodrošina vairākas tehnoloģijas – viedais līgums, konsensa mehānisms, šifrēšana, Merkla koks, jaucējfunkcijas, vienādranga tīkls. Šajā sadaļā tiks apskatītas dažas no šīm tehnoloģijām.

Visi dati, ar kuriem mēs mijiedarbojamies blokķēdē, tiek glabāti transakcijās. Transakciju dati nevar tikt mainīti, jo transakcijas tiek glabāti blokā Merkla koka veidā (sk. 2.1. attēlu). Viens

bloks var saturēt vairākas transakcijas, piemēram, Bitcoin bloks satur vidēji 1700 transakciju [22].

Katram blokam tiek pielietota jaucējfunkcija, kas jaucējvērtības aprēķināšanai izmanto arī iepriekšējā bloka jaucējvērtību. Tādējādi blokķēde ir bloku ķēde, kuru nekādā veidā nevar mainīt.

Zemāk redzamajā attēlā ir parādīta blokķēdes struktūra, taču jāņem vērā, ka metadatu kopums katrā atsevišķā blokķēdē var atšķirties.



2.1. attēls: **Blokķēdes struktūra**

Viedais līgums ir programma, kas darbojas decentralizētā infrastruktūrā, piemēram, blokķēdē. Tas ir drošs pret viltojumiem tādā nozīmē, ka neviena puse (pat to veidotājs) nevar mainīt to kodu vai traucēt to izpildi. Tādējādi blokķēdes lietotājs var būt drošs, ka nosacījums, kuram viņš piekrita, tiks izpildīts.

2.2. Konsenss mehānisms

Viena no lielākajām problēmām, ko risina blokķēde, ir atbrīvošanās no starpnieka. Mums vairs nav jāuzticas viens otram, lai būtu drošiem, ka visas puses pildīs savas saistības. Tagad to kontrolē konsenss mehānisms, kas nodrošina protokola noteikumu ievērošanu un visu transakciju

godīgumu un nodrošina, ka visi tīkla mezgli piekrīt jauna bloka pievienošanai. Konsensa algoritmam ir jānosaka, kurš pārbaudīs blokus un transakcijas tā, lai šis dalībnieks nekaitētu sistēmai. Lai to paveiktu, dalībnieki – maineri veic sarežģītus un/vai dārgus darbus un ar to pierāda, ka viņiem nav motivācijas kaitēt sistēmai. Tālāk ir īsi aprakstīti dažu algoritmu darbības principi:

- PoW (Proof-of-work) algoritms ir balstīts uz skaitļošanas jaudu. Tīkla mezgliem (angliski *nodes*), lai apstiprinātu transakcijas un neļautu citiem dalībniekiem divas reizes iztērēt vienas un tās pašas monētas, ir jāatrisina sarežģītas matemātiskas problēmas. Mezgls, kurš pirmais atrod risinājumu, tiek apbalvots, saņemot jaunas monētas. Šis algoritms ir diezgan drošs un, salīdzinot ar citiem algoritmiem, pastāv visilgāk un ir pārbaudīts laika ziņā. Taču tam ir arī daudz trūkumu, piemēram, tam ir ļoti augsts enerģijas patēriņš, augstas komisijas maksas, kā arī tas nav mērogojams un tam ir mazs transakciju pievienošanas ātrums.
- PoS (Proof of Stake) algoritmā jaunas monētas saņem nejauši izvēlēts tīkla dalībnieks. Izvēle notiek tikai starp tiem dalībniekiem, kuriem ir likme. Jo lielāka mezglam ir likme, jo lielāka iespēja, ka tas apstiprinās jaunu bloku un saņems atlīdzību.
- DPoS (Delegated Proof of Stake) algoritms ir uzlabota PoS (Proof of Stake) algoritma versija. PoS algoritms nodrošina, ka tiesības pievienot bloku tiks piešķirta mezglam, kuram ir lielāka likme. Savukārt DPoS algoritmam nav nepieciešama liela likme, šeit mezgli paši balso par mezglu, kurš izveidos bloku.
- PoSt (Proof of Spacetime) ir algoritms ar uzglabāšanas pierādījumu. Uzglabāšanas pierādījuma ideja ir pieprasīt tīkla dalībniekiem sagādāt tīklam krātuvi uz noteiktu laika periodu. Maineri pierāda, ka viņi fiziski glabā datus noteiktu laika periodu. Tas tiek pārbaudīts, nejauši atlasot mainerus un nolasot viņu datus pārbaudei.
- PoRep (Proof-of-Replication) ir algoritms, kas ļauj pierādīt, ka jebkura datu kopija tiek glabāta fiziski neatkarīgā krātuvē. Piemēram, Filecoin sistēmā mainerim ir jāpierāda lietotājam, ka dati ir replicēti savā unikālajā fiziskajā krātuvē.

Minētie PoS, PoW un DPoS konsensa algoritmi ir droši, jo krāpšanai nepieciešams 51% resursu no kopējā apjoma. PoW gadījumā tas ir 51% no tīkla skaitļošanas jaudas, PoS gadījumā 51% no visiem liktiem (angliski *staked*) marķieriem.

Tīkla mērogojamība nozīmē kā blokķēde ir spēcīga atbalstīt pieaugušo transakciju slodzi (cik daudz transakcijas tiek apstrādātas noteiktā perioda laikā) un palielināt mezglu skaitu tīklā. 2.2. tabulā norādīta minēto algoritmu mērogojamība.

Algoritms	**Projekti, kuros ir implementēts**	**Tīkla mērogojamība**
PoW	Bitcoin, Ethereum	Zema
PoS	Cardano, Algorand, Avalanche, Polkadot, Solana, kā arī Ethereum plāno pāriet uz PoS	Augsta
DPoS	EOS, BitShares	Augsta
PoRep	Filecoin	Zema
PoSt	Filecoin, DxChain	Augsta

2.2. tabula: **Populārāko konsensa algoritmu salīdzinājums**

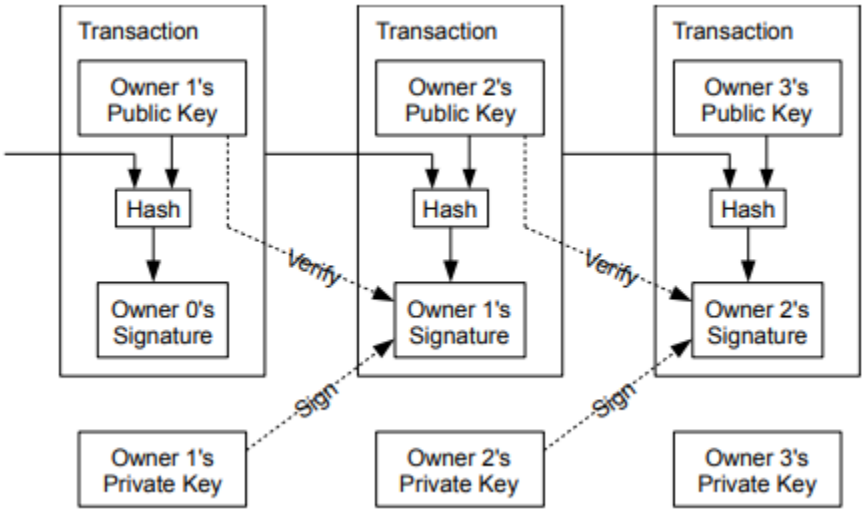
Konsensa algoritma mērogojamība ir svarīga Lielajiem datiem, jo datu ir daudz un tie pienāk pastāvīgi, dažreiz lielā ātrumā.

2.2.1. Šifrēšana

Blokķēdes veids (publiska, konsorcija vai privāta) neatrisina datu konfidencialitātes problēmu, ja transakcijas dati nav šifrēti, visa informācija par blokķēdi ir pieejama blokķēdes tīkla dalībniekiem, t.i. jebkurš mezgls, kam ir lasīšanas tiesības redzēs citu cilvēku privātos datus. Lai saglabātu cilvēku privātumu, datu šifrēšanai var izmantot simetrisku vai asimetrisku šifrēšanu pirms datu ievietošanas transakcijā. Parasti brīdī, kad lietotājs pievienojas, viņš izveido (ģenerē) privāto atslēgu, kuru saglabā pie sevis drošā vietā un ne ar vienu nedalās.

Kad lietotājs pievieno datus, viņš šifrē datus, izmantojot savu privāto atslēgu un simetrisku kriptogrāfiju (šifrēšanai un atšifrēšanai tiek izmantota viena un tā pati privāta atslēga, piemēram, DES, AES). Ja lietotājs vēlas nolasīt datus, viņš atšifrēšanai izmanto to pašu privāto atslēgu.

Piemēram, Bitcoin un Ethereum izmanto ECDSA (angliski *Elliptic Curve Digital Signature Algorithm*) asimetrisko algoritmu - privāta atslēga tiek izmantota digitālo parakstu izveidei un datu atšifrēšanai, publiskā atslēga tiek izmantota digitālo parakstu pārbaudei un datu šifrēšanai (sk. 2.2.attēlu).

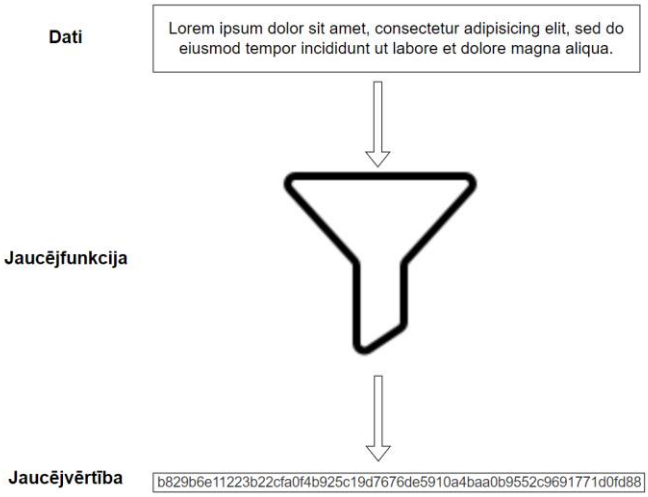


2.2.att: **Šifrēšanas algoritma darbības princips Bitcoin blokķēdē [14]**

Tādējādi blokķēdē ikviens, kuram ir lasīšanas tiesības, var iegūt bloku ķēdi, bet tikai datu īpašnieks var lasīt transakcijas datus - tas, kuram ir sākotnējā privātā atslēga.

2.2.2. Jaukšana

Jaucējfunkcijas (piemēram SHA-3, SHA-256) pārveido datus (sk. 2.3.attēlu) par jaucējvērtību (angliski *hash*). Jaucējvērtība ir noteiktā garuma virkne, no kuras nav iespējas dabūt atpakaļ sākotnējos datus. Blokķēdē jaucējfunkcija tiek izmantota, lai nodrošinātu ķēdes nemainīgumu: katram ķēdes blokam ir savs gandrīz unikāla (kolīzijas notiek ļoti reti) jaucējvērtība, kas veidojas no bloka datiem un iepriekšējā bloka jaucējvērtības (sk. 2.1.attēlu). Jaucējfunkcijas īpatnība ir tāda, ka ar jebkurām izmaiņām avota datos mainīsies arī jaucējvērtība.



2.3.att: **Jaucējfunkcijas darbības princips**

Tādējādi nav iespējams mainīt bloka datus, nemainot datus visā nākamajā ķēdē.

2.2.3. Merkla koks

Merkla koks ir binārs koks, tā lapu saknēs atrodas transakcijas (sk. 2.1.attēlu). Tālāk nākamajās virsotnēs glabājas jaucējvērtība, ko iegūst, piemērojot jaucējvērtību bērnu virsotņu vērtību summai. Ja trūkst kāda no virsotnēm, tad viena no virsotnēm tiek dublēta vai tiek pārnesta uz nākamo līmeni. Piemēram, mums ir Bitcoin blokķēdē tiek izmantota SHA-256 jaucējfunkcija.

Merkla koku izmanto, lai nodrošinātu, ka datu bloki, kas tiek nosūtīti caur tīklu, ir veseli, nebojāti un nemainīgi. Merkla koku var izmantot arī, lai izvairītos no vecu transakciju glabāšanas un ietaupītu vietu diskā [14]. Tā kā blokā tiek iekļauts tikai Merkla koka sākums, tad, ja pārējās daļas no Merkla koks pazūd, pats bloks netiks bojāts.

2.3. Blokķēdes paaudzes

Blokķēdes tehnoloģija nepārtraukti mainās un pielāgojas jaunām prasībām. Galvenās izmaiņas blokķēdes arhitektūrā iezīmē jaunas paaudzes sākumu. Šobrīd pastāv trīs veiksmīgi ieviestas blokķēdes paaudzes [9, 18, 21]:

2.3.1. Blokķēde 1.0

Šis ir senākais blokķēdes veids, to 2008. gadā aprakstīja anonīma persona vai personu grupa Satoši Nakamoto [14]. Attiecīgi pirmais projekts, kas ir balstīts uz blokķēdes 1.0 tehnoloģiju, ir Bitcoin.

Šajā blokķēdes versijā tika apvienotas vairākas tehnoloģijas un algoritmi: vienādranga tīkls (peer-to-peer network), konsensa mehānisms (kuru pirmo reizi aprakstīja S.Dvorka, N.Naors 1993. gadā [16]) un kriptogrāfija. Pirmā līmeņa blokķēdes izmanto PoW konsensa mehānismu, kas ir dārgs, jo prasa daudz elektroenerģijas. Bet šāda blokķēde ir piemērojama tikai naudas vai maksājumu sistēmām – digitālai valūtai.

2.3.2. Blokķēde 2.0

Otrā blokķēdes paaudze izmanto viedo līgumu. Pirmais projekts, kura pamatā izmantoja blokķēdi 2.0, bija Ethereum [17], kas tika palaists 2015. gadā.

Viedais līgums sniedz jaunas iespējas. Līdz ar viedo līgumu parādīšanos ir kļuvis iespējams veikt ne tikai parastus pirkšanas vai pārdošanas transakcijas, bet arī aprakstīt konkrētākas daudz sarežģītākas prasības, piemēram, obligācijas, hipotēkas, piegādes ķēdes, viedie īpašumi. Otrās paaudzes blokķēde ļāva veikt gan decentralizētas banku operācijas gan citus, ar naudu nesaistītus risinājumus, t.i., decentralizētu digitālo ekonomiku. Viedā līguma kods tiek izpildīts automātiski, kad ir izpildīti norādītie nosacījumi, tas ļauj lietotājiem paļauties, ka otra puse pildīs savas saistības, piemēram, samaksa par preci tiks veikta uzreiz pēc preces piegādes.

Tika uzlabots arī konsensa mehānisms, kas iepriekš prasīja ASIC iekārtu izmantošanu rakšanai. ASIC mikroshēmam ir liela skaitļošanas jauda, bet Ethereum izmanto DAG (Decentralized Acuclic Graphs), tas prasa zemu CPU un mazu atmiņu aprēķiniem. ASIC vietā Ethereum rakšanai izmanto grafikas apstrādes blokus (GPU).

Otrās paaudzes blokķēdes problēma paliek mērogošana, pārāk maz vietas blokķēdē datu (transakciju) glabāšanai, jo blokķēde joprojām izmanto PoW konsenss algoritmu un sadarbspēja (angliski interoperability) - iespēja redzēt un piekļūt informācijai dažādās blokķēdes sistēmās.

2.3.3. Blokķēde 3.0

Savukārt, trešā blokķēdes paaudze izmanto citus konsensa mehānismus, piemēram, PoS, DPoS, u.c., šādi algoritmi ļauj palielināt transakciju caurlaidspēju un attiecīgi, blokķēdes mērogojamību. Arī šajā versijā sāka izmantot starpķēžu (angliski cross-chain) tehnoloģiju. Starpķēžu tehnoloģija nodrošina informācijas apmaiņu starp blokķēdēm. Trešās paaudzes blokķēdes ir vairs nav izolētas, tās var veidot blokķēžu tīklu. Piemēram, Chainlink var nodrošināt datu apmaiņu starp vairākām blokķēdēm, šī tehnoloģija sīkāk ir aprakstīta sadaļā "Orākuli".

Populārākais projekts, kurā tiek izmantots blokķēde 3.0, ir Cardano. Cardano izmanto proof-of-stake (PoS) konsensa algoritmu, ko sauc par Ouroboros [19]. Cardano vietējais marķieris (angliski token) ir ADA. Šī projekta galvenā priekšrocība salīdzinājumā ar Ethereum ir tā, ka tas apstiprina blokķēdes transakcijas bez augstām enerģijas izmaksām.

Trešās paaudzes blokķēde sniedza iespēju veikt jaunus decentralizētus projektus, kas nav saistīti ar naudu, finansēm vai citām ekonomiskām darbībām. Šādi risinājumi ietver veselību,

izglītību, zinātni un citus sabiedriskos labumus, kultūras un komunikācijas aspektus. Trešā blokķēdes paaudze parasti ietver DApps (angliski Decentralized Apps).

Salīdzinot ar otrās paaudzes blokķēdi, blokķēde 3.0 ir palielinājusi ātrumu, mērogojamību un drošību.

2.3.4. Blokķēde 4.0

Blokķēde 4.0 šobrīd paliek teorija un vēl nav ieviesta, tāpēc šī paaudze šajā darbā netiks aplūkota.

2.3.5. Paaudžu salīdzinājums

2.3. tabulā ir redzams, ka trešās paaudzes blokķēdē ir ievērojami pieaudzis transakciju skaits sekundē (TPS). Transakciju ātrums galvenokārt ir atkarīgs no konsensa mehānisma, taču 1. un 2. paaudzē ar šīm vērtībām nepietiek, lai nodrošinātu netraucētu Lielo datu darbību. Un arī 3. paaudzē ir būtiski samazinājusies transakcijas cena, kas arī uzlabo apstākļus darbam ar Lieliem datiem.

Parametrs	Blokķēde 1.0	Blokķēde 2.0	Blokķēde 3.0
Pielietojums	digitāla valūta	digitāla ekonomika	digitāla sabiedrība
Jauna tehnoloģija	izplatīta virsgrāmata	viedais līgums	Ne-PoW konsenss algoritms
Implementācijas	Bitcoin, Dodgecoin, Litecoin	Ethereum	Cardano, IOTA, Polkadot
Transakcijas cena	1.5 USD [47]	0.69 USD [48]	0.16 USD [49]
Transakcijas ātrums	7 TPS	15 TPS	1000 TPS
Mērogojamība	Nav	Nav	Ir

2.3.tabula: **Salīdzinājums starp atšķirīgām blokķēdes paaudzēm**

Portālā Statista ir publicēts pētījums [20], kur tika apkopotas 66 populārākas kriptovalūtas un to vidējais transakciju apstiprināšanas laiks, ar kādu Kraken (kriptovalūtas birža un banka) apstiprinās noteiktu kriptovalūtu depozītu. Rezultātus var redzēt 2.4.tabulā zemāk.

Projekts	**Transakciju apstiprināšanas laiks**	**Blokķēdes paaudze**
Bitcoin	40 minūtes	1.0
Dodgecoin	40 minūtes	1.0
Litecoin	30 minūtes	1.0
Ethereum	5 minūtes	2.0
Cardano	10 minūtes	3.0
Polkadot	2 minūtes	3.0
Solana	5 sekundes	3.0
Ripple	5 sekundes	3.0

2.4.tabula: **Populārākie blokķēdes projekti un to transakciju apstiprināšanas laiks**

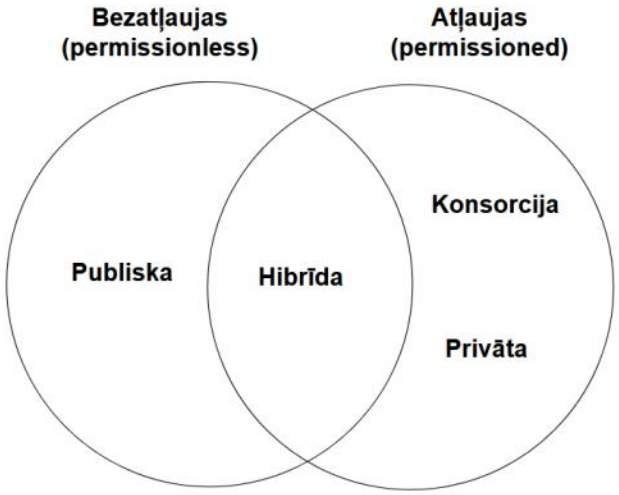
Vidējais transakcijas izpildes laiks svārstās no gandrīz 2-5 līdz pat vairākām stundām. Pirmās paaudzes kriptovalūtām šis laiks svārstās no 30 līdz 40 minūtēm, bet trešās blokķēdes paaudzes transakcijas var tikt apstrādāti gandrīz nekavējoties. Augstākais transakcijas ātrums nozīmē, ka blokķēde spēj pārsūtīt datus no vienas puses uz otru un apstiprināt transakcijas. Transakcijas ātrumu var ietekmēt vairāki faktori, tostarp bloka laiks, bloka lielums, transakcijas maksa un tīkla trafiks. Transakciju ātrumam ir svarīga loma decentralizētos Lielos datos.

Pirmā līmeņa blokķēdes tehnoloģija ir decentralizēta publiska virsgrāmata, ko galvenokārt izmanto valūtā un maksājumos, tā reprezentē digitālo valūtu. Otrā līmeņa blokķēdes tehnoloģija izmanto viedos līgumus un ļauj veikt sarežģītākus ekonomiskus darījumus. Otrās paaudzes blokķēdi bieži sauc par digitālo ekonomiku. Trešās paaudzes blokķēde izmanto jaunus konsensa algoritmus, kas paplašina blokķēdes mērogojamību, kas ļauj uzglabāt un apstrādāt lielu datu apjomu. Šo paaudzi dēvē par digitālo sabiedrību. 1. un 2. paaudzes blokķēžu problēma ir tāda, ka tās nav mērogojamas, tām ir vajadzīgas stundas ([21] ir minēts 60 minūtes), lai apstiprinātu transakcijas un tie patērē daudz enerģijas.

2.4. Blokķēdes veidi

Blokķēdes veids nosaka, kurš var lasīt, pievienot datus un piedalīties vienprātības mehānismā. Kā minēts [15], blokķēdes galvenokārt iedala 4 veidos (sk. 2.4. attēlu): publiskā,

privāta, konsorciju un hibrīda. Tie ir atdalīti pēc tā, vai jebkurš lietotājs var (bezatļaujas) vai nevar (atļaujas) lasīt, pievienot datus un piedalīties balsošanā.



2.4.att: **Blokķēdes veidi**

[15] pētījumā minētā tabula tika nedaudz modificēta un papildināta:

Parametrs	**Publiskā**	**Privāta**	**Konsorcija**	**Hibrīda**
Var lasīt	Ikviens	Iepriekš izvēlēta mezglu kopa	Iepriekš izvēlēta mezglu kopa	Tikai tie, kam ir atļauja
Var pievienot	Ikviens	Iepriekš izvēlēta mezglu kopa	Iepriekš izvēlēta mezglu kopa	Iepriekš izvēlēta mezglu kopa, vai tie, kurus pievienoja dalībnieki
Ātrums [21, 23]	Lēna	Ātra	Ātra	Atkarīgs no konfigurācijas
Kas var piedalīties konsenss mehānismā	Ikviens	Organizācija	Organizāciju grupas	Organizāciju grupas
Decentralizācijas līmenis	Pilnīgi decentralizēta	Centralizēta	Mazāk centralizēta	Drīzāk decentralizēta, nekā centralizēta

2.5.tabula: **Blokķēdes tipu salīdzinājums**

Tiesības pievienot datus un blokķēdes ātrums, ir atkarīgas no blokķēdes veida. Atvērtajiem datiem var izmantot publiska tipa blokķēdi, kas ir visvairāk decentralizētais blokķēdes veids.

No otras puses, ja mums jāpalielina transakciju ātrums, var izvēlēties "privātāku" blokķēdi, jo publiskā blokķēdē ikviens var pievienot transakciju, kas nozīmē, ka tīklā ir pārāk daudz pieprasījumu un transakciju ātrums palielinās. Privātajās blokķēdēs transakciju procesā var piedalīties tikai iepriekš izvēlētie mezgli. Tātad ātrums vienmēr paliek nemainīgs.

3. DECENTRALIZĒTI LIELIE DATI

3.1. Starpība starp centralizāciju un decentralizāciju un to ietekme uz Lieliem datiem

Centralizētā sistēmā visas darbības iet caur vienu grupu vai entītiju, kas nosaka šo darbību pareizību. Centralizētai sistēmai ir vairāki trūkumi, piemēram, šāds serveris ir vienots atteices punkts (angliski *single point of failure*), sistēma ir vairāk pakļauta DDoS uzbrukumiem, kā arī kļūst nepieciešams paļauties uz servera/organizācijas lēmumiem.

Decentralizētā sistēmā tiek izmantots P2P tīkls, kura dalībnieki ar balsu vairākumu nobalso par lēmuma pieņemšanu.

Glabāt Lielo datu informāciju centralizēti var būt neizdevīgi, jo pastāv liela datu uzbrukuma iespējamība, izmaksas par krātuvēm, iespēja pietiekami ātri paplašināt un atjaunināt sistēmu, lai apmierinātu lietotāju pieprasījumu pēc ātrākiem datiem un lielākiem formātiem. Populārākie centralizēto Lielo datu glabāšanas piemēri ir Apache Hadoop, Amazon Simple Storage Service.

Decentralizēta krātuve (Decentralized Storage Network - DSN) izmanto blokķēdes tehnoloģiju krātuves pārvaldībai. Kā jau aplūkots iepriekšējā sadaļā, trešās paaudzes blokķēdei ir visi nepieciešamie parametri Lielo datu pārvaldībai un uzglabāšanai – liels transakciju ātrums un zemas izmaksas, mērogošanas iespēja. Decentralizētas datu uzglabāšanas piemēri ir IPFS [43], Filecoin [44], Storj [45], DxChain [39].

3.2. Kur glabāt datus?

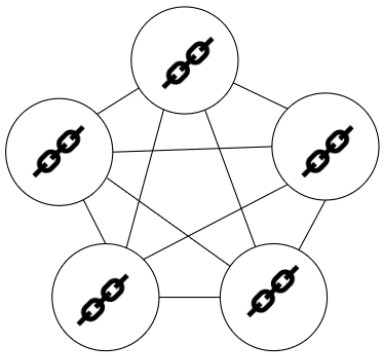
Līdz ar trešās paaudzes blokķēdes parādīšanos mums ir iespēja uzglabāt datus lielā apjomā, ir palielinājies transakciju apstrādes ātrums (sk. nodaļu "Blokķēdes paaudzes"). Trešās paaudzes blokķēdi var izmantot liela mēroga datu pārvaldībai, piemēram, elektroniskās medicīniskās kartes sistēmā.

Saskaņā ar [28], Lielo datu apjoms, ko ģenerē dažādas organizācijas, tiek aprēķināts petabaitos (PB) un eksabaitos (EB). Attiecīgi mums ir nepieciešams risinājums, kas spēj uzglabāt

un apstrādāt vismaz eksabaitus informācijas, un, ņemot vērā, ka datu apjoms nepārtraukti pieaug, zetabaitus informācijas.

3.2.1. Tehniskie ierobežojumi

Blokķēdes tīkls ir vienādranga (P2P) tīkls, kurā katrs pilnais mezgls glabā visas virsgrāmatas (bloku ķēdes) kopiju (sk. 3.1. attēlu). Blokķēdes, piemēram, Bitcoin vai Ethereum, kas savā darbā izmanto PoW konsensa mehānismu, nespēj uzglabāt milzīgu apjoma datus, to mērogojamību dēļ. Saskaņā ar Statista datiem uz 2022.gada 4. aprīli [34] Bitcoin blokķēdes virsgrāmatas izmērs ir 389,72 GB. Savukārt Ethereum virsgrāmatas izmērs uz 2022.gada maijam ir 500 GB – 1 TB [35], Ja ķēde tiktu paplašināta līdz lielam datu apjomam (teiksim 5 TB) ne visi mezgli varēs turpināt darboties. Un arī transakcijas maksas cena būs ļoti augsta [36].



3.1.att: **Blokķēdes P2P tīkls**

Saskaņā ar Vitalika Buterina (Ethereum dibinātājs) rakstu [37], galvenie ierobežojumi kas neļauj pilnam mezglam apstrādāt lielu skaitu transakciju ir šādi:

- **Skaitļošanas jauda: cik % no CPU mēs varam droši pieprasīt, lai palaistu mezglu?**

  Tikai ~5-10% no CPU var iztērēt bloka pārbaudei, jo (1) mums ir nepieciešama CPU rezerve DDoS uzbrukuma gadījumā, (2) gadījumā ja tīkla savienojums pazuda, pēc tām, kad savienojums tiks atjaunots, mezglam jāspēj atjaunināt virsgrāmatu un sinhronizēties ar pārējiem mezgliem dažu sekunžu laikā, (3) mezglam nevajadzētu pārtraukt citu lietojumprogrammu darbu tajā pašā mezglā, (4) papildus bloku validācijai mezgls veic arī citus uzdevumus, kas saistīti ar blokķēdes darbības atbalstu, piemēram, transakcijas validāciju.

- **Caurlaidspēja (datu pārraide - cik ātri varam sūtīt datus): ņemot vērā pašreizējo interneta savienojumu realitāti, cik baitu var saturēt bloks?**
  Ņemot vērā to, ka Interneta pakalpojumu sniedzēju reklamētais ātrums ne vienmēr atbilst realitātei un to, ka mezgli bieži lejupielādē transakcijas vairākas reizes, nevajadzētu paļauties uz vidējo caurlaidspēju ātrumu. Maksimālais ātrums, ko [37] rekomendē ir 1-5 MB bloki ik pēc 12 sekundēm.
- **Krātuve: cik gigabaitu diskā var pieprasīt lietotājiem saglabāt un cik ātri tam jābūt lasāmam?**
  Teorētiski Amazon var viegli iegādāties 15 TB SSD disku [38], taču reāli parastie cilvēki izmanto parastu datoru, kurā pieejamā SSD diska vidēja ietilpība ir 512 GB [37].

Ja mēģināt saglabāt Lielos datus blokķēdes virsgrāmatā, tad šajā gadījumā katram pilnajam tīkla mezglam būs jāuzglabā milzīgs datu apjoms un kādā brīdī diska vieta vienkārši beigsies.

Modernās decentralizētās krātuves pārņem arhitektūru no esošajiem risinājumiem un pievieno tam blokķēdes līmeni.

3.2.2. IPFS (Interplanetary File System)

IPFS ir izplatīta sistēma (IPFS nav blokķēde) failu, vietņu, lietojumprogrammu un datu glabāšanai un piekļuvei tiem. IPFS nodrošina augstas caurlaidspējas bloku krātuves modeli, kā arī nodrošina versiju kontroles funkcionalitāti [43]. IPFS pārņēma BitTorrent arhitektūru.

IPFS visi mezgli ir publiski, mezgli glabā IPFS objektus lokālajā krātuvē. Mezgli var savienoties viens ar otru un sūtīt objektus. Šie objekti attēlo failus un citas datu struktūras. Katrs fails, ko saglabā lietotājs, IPFS sistēmā tiek sadalīts mazākos noteiktā izmēra gabalos tos sauc par *blokiem*.

IPFS arhitektūru var redzēt 3.2.attēlā. IPFS struktūra sastāv no vienādranga tīkla, Merkla kokas un DHT tabulām.

Merkla koks

Merkla koka lapas satur bloka jaucējvērtību, bet Merkla koka saknes jaucējvērtība tiek izmantota, lai identificētu failu un veido satura identifikatoru (angliski *content identifier* CID).

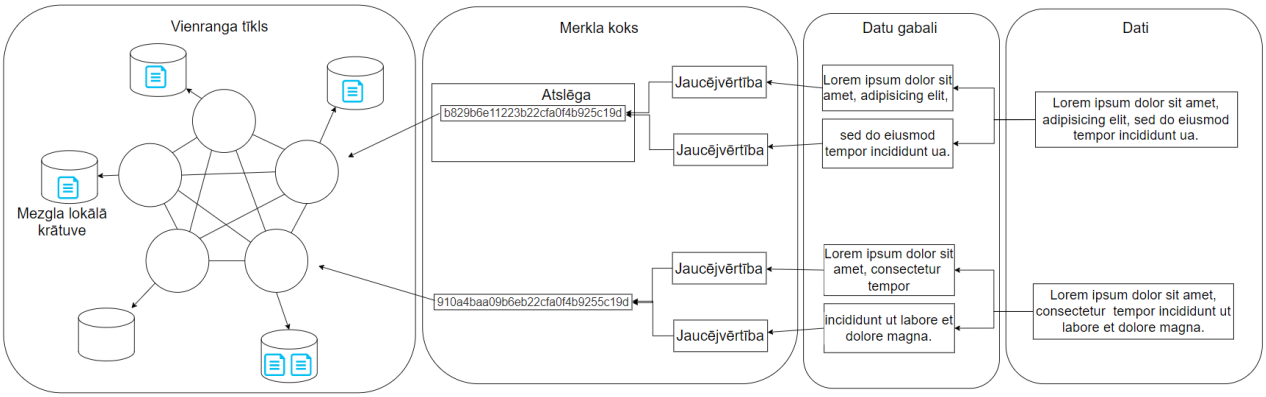
CID sastāv no Merkla koka saknes jaucējvērtības, IPFS versijas, kodēšanas metodes (piemēram, sha2), prefiksa un satura formāta, kas ir nepieciešams, lai programmas zinātu, kā interpretēt faila saņemto saturu (fails, mape, repozitorijs). CID ir adrese, ko IPFS izmanto, lai norādītu uz materiālu.

Merkla koka nodrošina arī versiju kontroli. Kad fails tiek mainīts, Merkla koks atjaunina tikai atsauces vai pievieno jaunus objektus kokā. Šajā gadījumā faila CID mainīsies, tas nozīmē, ka lietotājam joprojām ir piekļuve vecajai faila versijai un arī jaunajai. Pēc izmaiņām visi mezgli atjaunina savas DHT tabulas, pievienojot jaunas CID vērtības.

Izplatītās jaucējtabulas

DHT (angliski *Distributed Hash Table* DHT) ir decentralizēta krātuves sistēma, kas nodrošina datu meklēšanu un saglabāšanu, izmantojot <*key, value*> pārus. DHT darbojas kā kataloga un navigācijas sistēmas krustojums. DHT saista to, ko lietotājs meklē, ar mezglu, kas glabā atbilstošo saturu, t.i., <key = CID, value = mezgla ID> vai < key = mezgla ID, value = [CID1, CID2]>.

Katrs IPFS mezgls satur DHT maršrutēšanas tabulu ar saitēm uz citiem mezgliem tīklā.



3.2.att: **IPFS arthitektūra**

3.1.tabulā ir aprakstīta faila pievienošanas un izgūšanas procedūra. Aprakstītie lietošanas gadījumi ir nepilnīgi un satur tikai galvenos punktus.

Faila pievienošana	**Faila iegūšana**
1. Lietotājs pievieno failu IPFS. 2. Fails tiek sadalīts mazākos blokos. 3. Katrs bloks ir ievietots Merkla kokā. 4. Pēdēja jaucējvērtība Merkla koka saknē, IPFS versija, kodēšanas, prefikss un satura formāts, veido CID. 5. Fails tiek saglabāts lokālajā krātuvē. 6. CID un mezgla ID tiek pievienoti DHT tabulai un tiek izplatīti pa tīklu. 7. Citi mezgli pieprasa failu pēc CIT un saglabā to pie sevis.	1. Lietotājs pieprasa noteiktu failu un norāda faila atslēgu (CID). 2. Mezgls (1) meklē savā DHT tabulā attiecīgo CID un mezgla ID, kurš glabā failu. 3. Mezgls (1) pieprasa IP adreses maršrutu, lai nokļūtu norādītajā ID mezglā. 4. Mezgls (1) izveido savienojumu ar norādīto mezglu (2) un pieprasa CID, ko vēlas iegūt. 5. Mezgls (2) nosūta mezglam (1) norādītus blokus. 6. Pēc bloka izgūšanas IPFS pārbauda, vai bloku un satura jaucējvērtības sakrīt (tādējādi IPFS pārbauda, vai pārsūtīšanas laikā dati nav mainīti). 7. Faila dati tiek saglabāti mezgla (1) kešatmiņā un kļūst pieejami citiem mezgliem, līdz tiek notīrīta kešatmiņa.

3.1.tabula: **IPFS darbu plūsma**

Pamatojoties uz IPFS tehnoloģiju, tika izveidoti vairāki blokķēdes projekti, kas ir saistīti ar Lielo datu glabāšanu un apstrādi, piemēram, Filecoin, OrbitDB, Origin. IPFS struktūra nodrošina liela apjoma failu glabāšanu izplatītā veidā.

IPFS nodrošina failu nemainīgumu, jo izmanto Merkla kokas un jaucējfunkcijas. Ja kāds mainīs faila vai bloka saturu, mainīsies arī jaucējvērtības, tādā veidā var pārbaudīt faila oriģinalitāti. Jaucējvērtības arī nodrošina faila unikalitātes pārbaudi - ja lietotājs augšupielādē failu, kas jau atrodas sistēmā, tad failiem būs viena un tā pati CID.

Pieņemot, ka katrs lietotājs iegulda 500 GB krātuves, 5 ZB krātuvei būtu nepieciešami 5 000 000 mezglu. Filecoin atrisina šo problēmu, palielinot prasības ierīces parametriem, tāpēc par IPFS trūkumu var uzskaitīt to, ka efektīvai IPFS darbībai ir nepieciešams liels mezglu skaits.

3.2.3. Filecoin

Filecoin [44] darbojas kā slānis virs IPFS, kas var nodrošināt jebkuru datu uzglabāšanas infrastruktūru, tā blokķēde darbojas, izmantojot *proof-of-storage* konsensa mehānismu. Filecoin darbībai izmanto marķieri, ko sauc par "Filecoin" (FIL). Blokus blokķēdē veido maineri, kas glabā datus

Protokols

Filecoin protokols nodrošina datu glabāšanas un ieguves pakalpojumu. Blokķēdes maineri izveido blokus un glabā datus, saņemot par to naudu. Klienti maksā par datu glabāšanu un izgūšanu.

Protokols iedala klienta pieprasījumus 2 daļās: tie, kas saistīti ar uzglabāšanu (Storage Market SM), un tie, kas saistīti ar izguvi (Retrieval Market RM). SM un RM pārvalda Filecoin tīkls, kas izmanto *proof-of-storage*, lai garantētu, ka maineri ir pareizi saglabājuši datus. Savukārt, maineri piedalās bloku izveidē, varbūtība, ka maineris saņems iespēju izveidot bloku, ir proporcionāls viņu tīklā izmantotās krātuves apjomam.

Filecoin ir sekojošas aparatūras prasības krātuves nodrošinātājiem:

- 8+ kodolu centrālais procesors
- Vismaz 138 GB RAM
- Jaudīgs GPU
- ~1 TB uz NVMe balstīta diska vieta kešatmiņas glabāšanai
- Papildu cietie diski slēgto sektoru un citu glabāšanai

Kā var redzēt, ierīcēm ir diezgan augstas prasības, īpaši 138 GB RAM.

Proof-of-Storage

Proof-of-Storage sastāv no diviem algoritmiem - Proof-of-Replication (PoRep) un Proof-of-Spacetime (PoSt), un tas ir vajadzīgs maineriem, lai pierādītu klientiem, ka viņu dati tiešām tiek glabāti mainera krātuvē.

PoRep (sk. sadaļā "Konsenss mehānisms") var palīdzēt izvairīties no šādām problēmām:

- Ļaunprātīgi maineri varētu teikt, ka viņi uzglabā (un saņem par to samaksu) vairāk kopiju, nekā viņam ir.

- Ļaunprātīgi maineri varētu apņemties uzglabāt vairāk datu, nekā viņi var fiziski uzglabāt.
- Ļaunprātīgi maineri var apgalvot, ka glabā lielu datu apjomu, un tas palielina ļaunprātīgā kalnraču iespējamību dabūt atlīdzību par bloka izveidošanu

Jebkurš lietotājs var piedalīties kā klients vai maineris, kurš glabā un/vai iegūst datus, t.i., Filecoin ir publiskā veida blokķēde. Klientiem, kuri vēlas, lai viņu dati tiktu glabāti privāti, ir jāšifrē datus pirms to iesniegšanas tīklā.

Filecoin var kalpot kā datu krātuve Lielajiem datiem, viņš darbojas atvērtā vienādranga tīklā, vienlaikus nodrošinot ekonomiskus stimulus (atlīdzības) un pierādījumus, lai nodrošinātu failu pareizu glabāšanu.

3.2.4. DxChain

DxChain projekts [39] ir paredzēts, lai kalpotu kā datu tirdzniecības platforma lietotājiem, kuri vēlas pārdot savus datus, tas nodrošina datu uzglabāšanu un elastīgu failu izgūšanu. Šis tīkls ļauj uzglabāt lielus datus un izmanto mašīnmācīšanās aprēķinus, lai apstrādātu datus. DxChain arhitektūra ir balstīta uz IPFS, HDFS un citiem projektiem. DxChain izmantoja Lielo datu apstrādes platformu Hadoop kā piemēru datu apstrādei.

Lai nodrošinātu sistēmas darbību, DxChain izmanto iekšējo marķieri, ko sauc par DX. DxChain arhitektūra sastāv no galvenās ķēdes un divām sānu ķēdēm (angliski *side chain*):

Galvenā ķēde (angliski *master chain* MC)

Ķēde pārvalda uzglabāšanas ķēdi un skaitļošanas ķēdi. MC izmanto datu struktūru, kas ir savietojama ar Ethereum. MC ir virsgrāmata, kurā glabājas informācija par pabeigtām transakcijām un metadatiem, savukārt sarežģītas datu struktūras un skaitļošanas informācija glabājas pārējās sānu ķēdēs (side chains). MC izmanto DPoS konsensa algoritmu.

Datu sānu ķēde (angliski *data side chain* DSC)

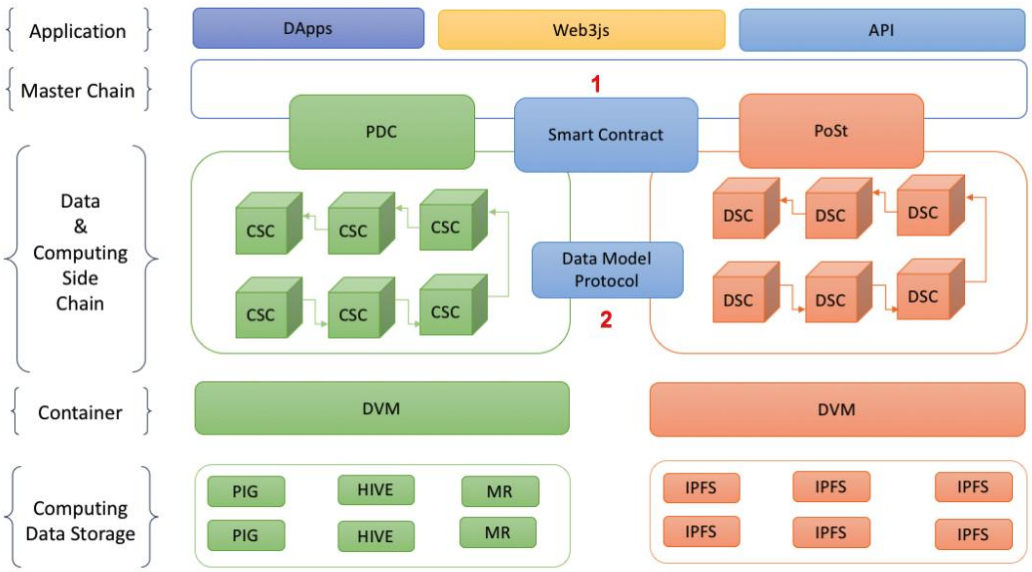
Paredzēta Lielo informāciju glabāšanai un balstīta uz P2P izplatīto failu glabāšanas sistēmu (līdzīgs HDFS, sk. “Apache Hadoop” sadaļu). Izmanto PoSt algoritmu, lai pārbaudītu, vai tiešām lietotājs glabā datus. Dati un faili ir sadalīti mazās daļās un tiek glabāti vienādranga krātuves tīklā (kā piemēram, IPFS). Datu vai failu jaucējvērtība glabājas transakcijās Merkla koka veidā.

Skaitļošanas sānu ķēde (angliski *computing side chain* CSC)

Lai nepārslogotu galveno ķēdi ar skaitļošanu, kas prasa daudz resursu, visi aprēķini (MapReduce vai datubāzes vaicājumi) notiek CSC. CSC ir paredzēta, lai atrisinātu biznesa problēmas, tā atbalsta konkrētu skaitļošanas uzdevumu DxChain virtuālajā mašīnā (DVM). CSC var nolasīt datus no DSC un ierakstīt rezultātu atpakaļ uz DSC. CC izmanto divus mehānismus, lai nodrošinātu aprēķinu pareizību: Verification Game ietvaru un Provable Data Computation (PDC) paradigmu. Verificaton Game ietvars aktivizē viedos līgumus, lai veiktu jebkuru skaitļošanas uzdevumu. PDC ir algoritms, kurš var pierādīt jebkura aprēķina pareizību. DxChain arī piedāvā Hadoop MapReduce (sk. sadaļā "Apache Hadoop") implementāciju datu apstrādei.

Komunikācija

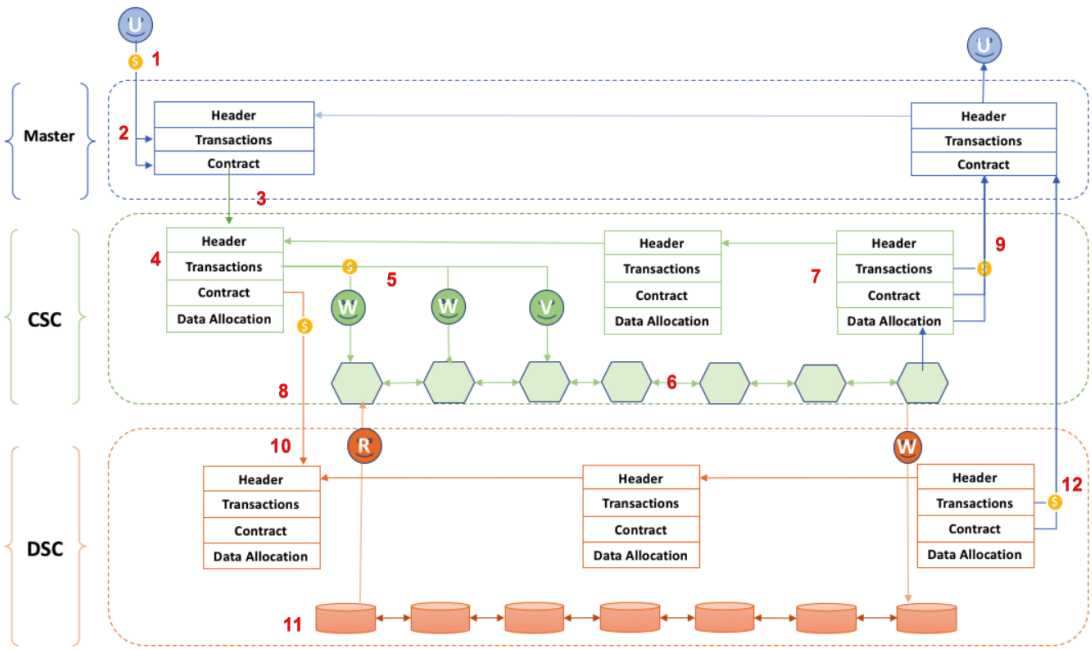
DSC un CSC sazinās ar galveno ķēdi, izmantojot DxChain tīkla viedo līgumu (1) (sk. 3.3.attēlu). DSC un CSC var arī savstarpēji sadarboties, izmantojot Data Model Protocol (2). DC un CC apstrādā datus tikai noteiktu laiku, ja datiem ir beidzies derīguma termiņš, tad datus vairs nevajag glabāt ķēdēs. Visas trīs ķēdes ir pilnīgi izolētas, piemēram, galvenā ķēde netiek ietekmēta pat tad, ja sānu ķēdes ir uzlauztas.



3.3.att: **DxChain arhitektūra [39]**

Tālāk ir secīgi aprakstītas darbības, ko veic DxChain tīkls, kad lietotājs iesniedz vai pieprasa datus (arī ir parādīts 3.4.attēlā):

1) Lietotājs (U) iesniedz uzdevumu MC.
2) MC pārbauda datu pareizību, izveido transakciju ar datiem par uzdevumu.
3) MC nodod datus CSC.
4) CSC saņem vaicājumu, vaicājums tiek izplatīts tīklā starp mezgliem. Nepieciešamības gadījumā lasa/raksta datus no DSC.
5) Verificaton Game ietvars ielādē datus DxChain virtuālajā mašīnā (DVM).
6) DVM izpilda aprēķinus, tostarp paralēlo skaitļošanu un pārbaudes uzdevumus.
7) Verification Game ietvars pārbauda aprēķina rezultātu, CSC mezgls atjauno uzdevuma statusu CSC blokķēdē.
8) CSC nodod datus DSC.
9) Nodod informāciju par noteikto darbību MC un saņem atlīdzību.
10) DSC saņem vaicājumu datu saglabāšanai vai iegūšanai, tālāk šis vaicājums tiek izplatīts tīklā starp mezgliem.
11) Izvēlētais mezgls saglabā vai iegūst datus no izplātītas krātuves, atgriež datus tam, kurš pieprasīja datus.
12) Atjaunina DSC blokķēdi, pievienojot jaunu transakciju un nodod informāciju par noteikto darbību MC un saņem atlīdzību.



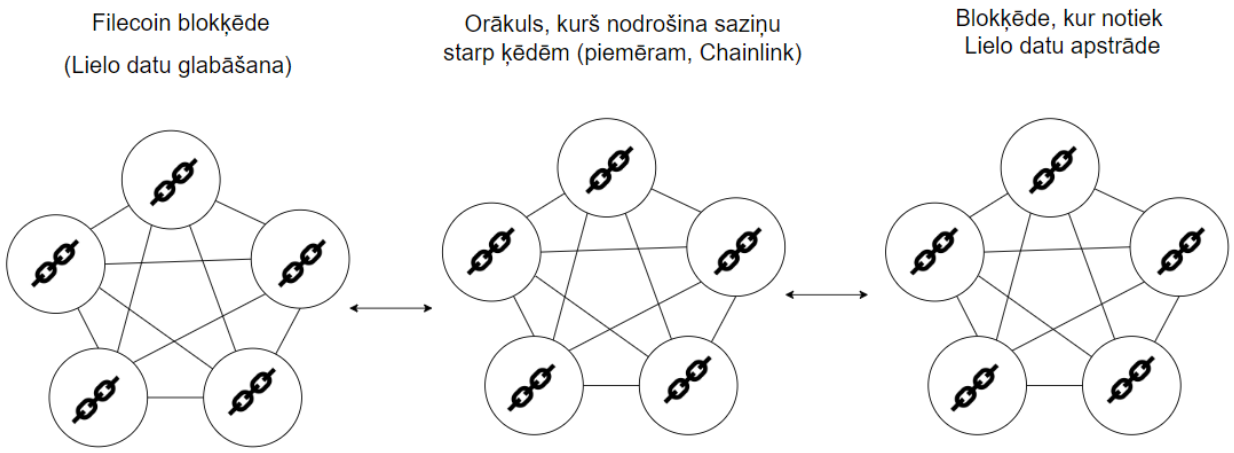
3.4.att: **DxChain darba plūsma [39]**

DxChain ir sava veida decentralizēts Hadoop, kur par *NameNode* kalpo galvenā ķēde (*master chain*). Šis projekts joprojām ir izstrādes stadijā, taču izstrādātāji jau ir palaiduši galvenās ķēdes beta versiju.

Kopsavilkums

Lielo datu uzglabāšanas problēmas risināšanai ir vairāki risinājumi: (1) izplatīta failu sistēma (līdzīgi ka HDFS) un (2) IPFS piedāvāto risinājumu, kas ir vienādranga tīkla, DHT un Merkla koka tehnoloģiju kombinācija. Lielākā daļa risinājumu izmanto IPFS sistēmu.

Lielo datu apstrādes un glabāšanas blokķēdes sistēmas izveide no jauna var būt sarežģīta, dārga un laikietilpīga. Viena no vienkāršākajām iespējām lielu datu glabāšanai ir izmantot Filecoin tīklu datu glabāšanai, blokķēdes var būt saistītas ar orākula palīdzību (sk. 3.5. attēlu).



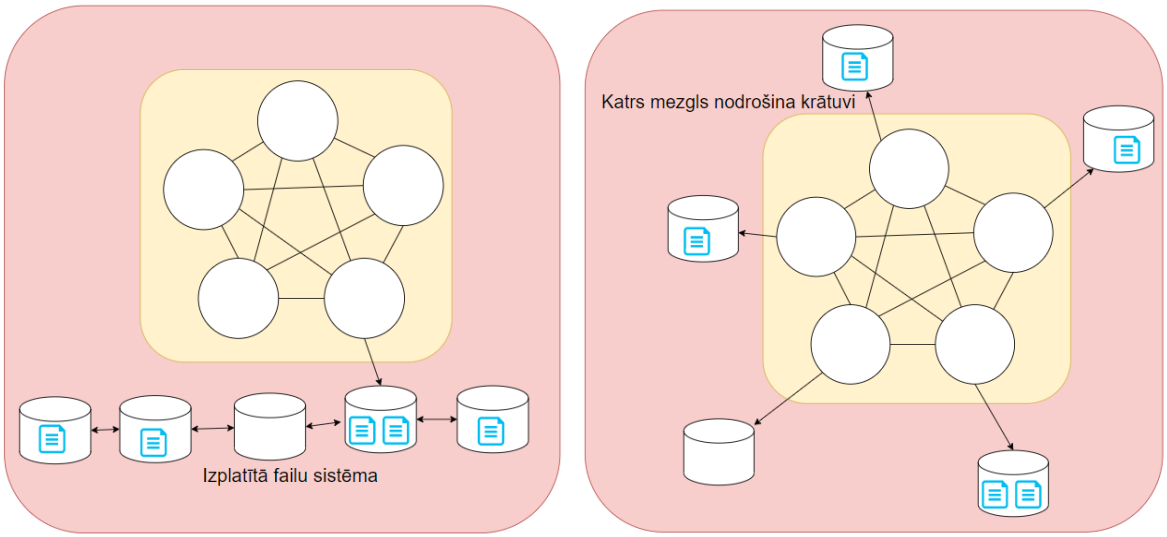
3.5.att: **Blokķēdes risinājums Lielu datu apstrādei un glabāšanai**

Lielo datu apstrādei DxChain plāno izmantot trīs neatkarīgus blokķēdes tīklus, no kuriem katrs tīkls pildīs savu darbu: uzdevumu koordinēšanu, aprēķinus un datu glabāšanu. Jāpatur prātā, ka, salīdzinot ar IPFS un Filecoin, DxChain platforma vēl nav pierādījusi savu veiktspēju, jo šī darba rakstīšanas laikā tā ir daļēji palaista beta versijā. Pret šī projekta, kā arī citiem autora piedāvātiem risinājumiem jāizturas kritiski.

3.2.5. Off-chain un On-chain storage

Datus var uzglābāt gan blokķēdē, gan ārpus tās. Blokķēdē parasti tiek glabāti tikai konkrēti datu elementi (piemēram, transakcijas, metadati). Lielāka daļa risinājumu apvieno blokķēdi ar ārpusķēdes krātuvi, piemēram Storj, Filecoin.

3.6.attēlā dzeltenais kvadrāts parāda on-chain krātuvi, bet sarkans – off-chein.



3.6.att: **Decentralizētas krātuves arhitektūra**

3.3. Orākuli decentralizētos Lielos datos

Bieži vien blokķēdes sistēmai, lai veiktu darbības, ir jābūt pieejamai informācijai reālajā pasaulē, piemēram, informācija par laikapstākļiem noteiktā reģionā. Bet, ja blokķēde uzticas informācijai tikai no viena avota, tad pastāv iespēja, ka šis avots tiks uzlauzts vai arī tas sniegs nepareizus datus.

Ja mēs vēlamies ievērot galveno blokķēdes filozofiju – decentralizāciju, tad nevaram ņemt datus no vienas vietas un uzticēties tikai vienam datu avotam. Lai atrisinātu šo blokķēdes datu ieguves problēmu, tika izgudroti orākuli.

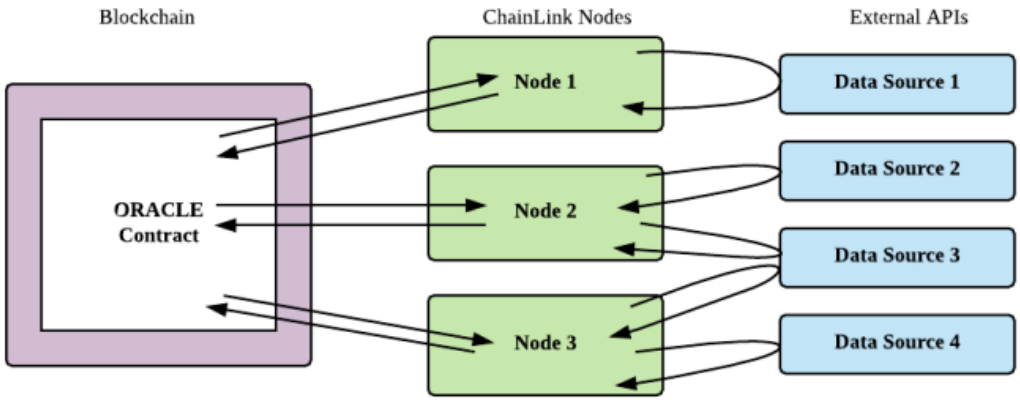
Orākuli savieno blokķēdi ar ārējām sistēmām. Nodrošina ievadi un izvadi reālajai pasaulei [2]. Chainlink ir viens no vispazīstamākajiem decentralizētiem orākula tīkliem (DON). Tālāk tiks aplūkota orākula arhitektūra, izmantojot Chainlink piemēru [31].

3.3.1. Arhitektūra

Informācija, ko sniedz centralizēts avots, var nebūt patiesa, tāpēc decentralizēts orākuls nevar paļauties tikai uz vienu avotu. Vienkāršs veids, kā rīkoties ar nepareizu vienu avotu, ir iegūt datus no vairākiem avotiem. Decentralizēts orākuls var pieprasīt datus no vairākiem avotiem $Src_1, Src_2, \dots, Src_k$, iegūt atbildes $a_1, a_2, \dots, a_k$ un apkopot tās vienā atbildē $A = agg(a_1, a_2, \dots, a_k)$. Tālāk pareizā rezultāta noteikšanas algoritms var atšķirties, piemēram, *agg* funkcija var noteikt pareizo rezultātu ar balsu vairākumu, t.i. vairāk nekā *k/2* (ja vairumā avotu pareizā atbilde ir *a*, tad rezultāts būs *a*), vai, piemēram, funkcija *agg* var atgriezt rezultātu vidējo vērtību, atmetot novirzes (lielākās un mazākās vērtības a_i), vai arī citas pieejas. Bet jāatceras, ka kļūdas avotos joprojām ir iespējamas, it īpaši, ja avoti atsaucas viens uz otru un rezultāts šādos gadījumos var nebūt pilnīgi patiess [33].

Papildus tam, ka ir jāizplata datu avoti, ir jāizplata arī pats orākuls, pretējā gadījumā tas būs vienots atteices punkts (angliski *single point of failure*). Orākulam var būt daudz mezglu $\{O_1, O_2, \dots, O_n\}$. Katrs orākula mezgls O_i sazinās ar savu atšķirīgo datu avotu kopu, kas var pārklāties vai arī var nepārklāties ar citu orākulu datu avotu kopu (sk. 3.7.attēlu). Katrā mezglā O_i datus apstrādā *agg* funkcija un tik saņemts rezultāts A_i. Iegūtais rezultāts A*i* var atšķirties katra mezgla gadījumā, jo tie apstrādā dažādus avotus. Tālāk saņemtos datus $\{A_1, A_2, \dots, A_n\}$ var apstrādāt dažādos veidos. Piemēram, var izmantot orākula līgumu (Oracle contract), kas savukārt apkopo datus $A=Agg(A_1, A_2, \dots, A_n)$ un sūta saņemto atbildi lietotājam, kurš pieprasīja informāciju. Bet šajā risinājumā par O(n) orākula ziņojumu pārraidīšanai un apstrādei ir jāmaksā komisijas. [33] apgalvo, ka tādas izmaksas atļaujas (permissioned) blokķēdēm var būt pieņemams, bet ir diezgan dārgs bezatļaujas (permissionless) blokķēdēm. Tāpēc Chainlink savā risinājumā izmanto sliekšņa parakstus (TS). TS ir elektroniskā paraksta variants, kam nepieciešams, lai sadarbotos vismaz *t* dalībnieku skaits no *n*. Piemēram, mums ir O_n mezgli. Pirmkārt, tiek ģenerēta privātā atslēga. Tālāk privātā atslēga tiek sadalīta *n* daļās un sadalīta starp *n* mezgliem, katrai privātajai atslēgai tiek ģenerēta publiskā atslēga. Rezultātā katram mezglam ir savs unikāls atslēgu pāris (pk_i, sk_i). Tālāk mezgls ģenerē parakstu $\sigma_i = Sig_{sk_i}[A_i]$, ko var pārbaudīt ar pk_i. Galīgas atbildes $A=Agg(\sigma_1, \sigma_2, \dots, \sigma_k)$ izveidē jāpiedalās vismaz *t* mezgliem. Rezultātā galīgā atslēga tiek aprēķināta, pamatojoties uz *A* rezultātu: $\Sigma = Sig_{sk}[A]$. Atslēga Σ būs derīga tikai tad, ja atslēgas izveides procesā ir piedalījušies vismaz *t* mezgli [33].

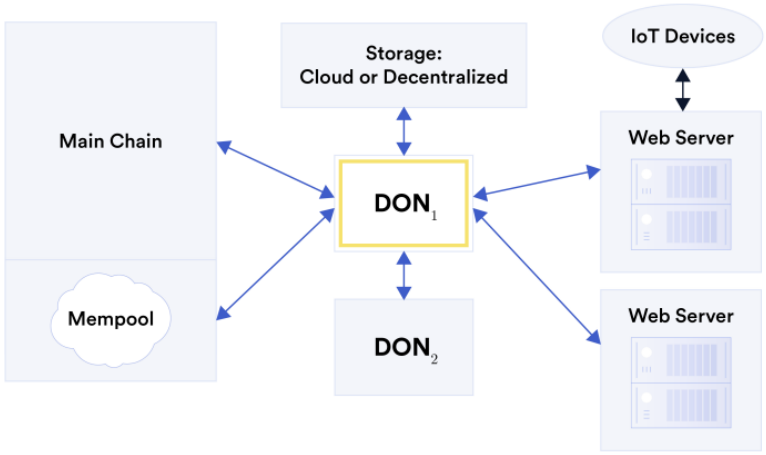
Katrs neatkarīgais mezgls decentralizētajā orākula tīklā neatkarīgi iegūst datus no ārēja avota, tādējādi nodrošinot kvalitatīvu datu piegādi.



3.7.att: **Orākula divu līmeņu izplatīšana [33]**

Chainlink orākuli izmanto saskarnes, caur kurām izpildāmie faili, kas darbojas uz DON, var sūtīt un saņemt datus no ārpus-DON sistēmām. Chainlink publicēja [31] balto papīru (angliski *whitepaper*) 2.0 versijai, kurā adapteri var tikt izveidoti šādiem ārējiem resursiem (parādīts 3.8.attēlā):

- Blokķēdes (attēlā *main chain* un *mempool*). Adapteris var definēt, kā sūtīt transakcijas uz blokķēdi un kā no tās nolasīt blokus vai transakcijas. Adapteri var strādāt arī ar blokķēdes *mempool*.
- Tīmekļa serveri (attēlā *web server*). Adapteri var definēt API, caur kurām var izgūt un sūtīt datus uz/no tīmekļa serveriem. Tīmekļa serveri, ar kuriem savienojas DON, var kalpot kā savienotājs ar papildu resursiem, piemēram, IoT ierīcēm.
- Ārējā krātuve: adapteris var definēt metodes lasīšanai un rakstīšanai uzglabāšanas pakalpojumos ārpus DON, piemēram, decentralizētā failu sistēmā vai mākoņkrātuvē.
- Citi DON: Adapteri var izgūt un pārsūtīt datus starp DON.

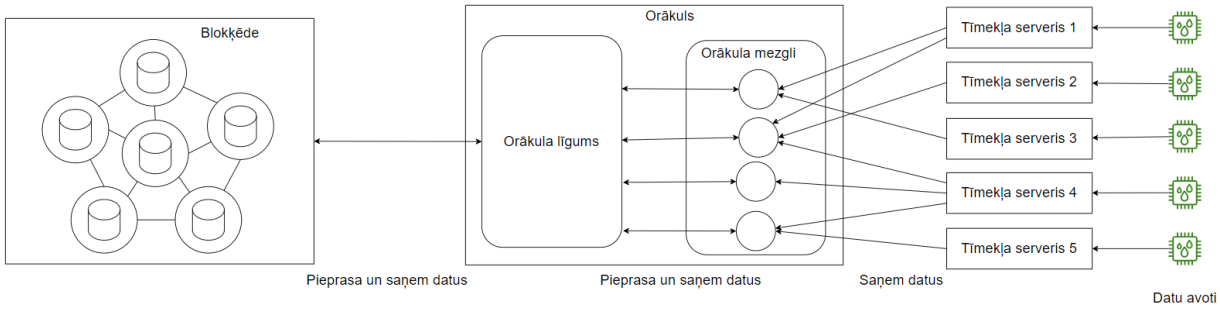


3.8.att: **Orākula arhitektūra [31]**

Šāda DON funkcionalitāte var būt noderīga DLD, kas saņemtu datus no ārējiem avotiem, tīmekļa serveriem vai mijiedarbotos ar citām sistēmām.

3.3.2. Kopsavilkums

Decentralizētus orākulus var izmantot kvalitatīvu datu iegūšanai, apkopojot datus no vairākiem avotiem un nosakot patiesāko rezultātu. 3.9.attēlā orākuls kalpo kā starpnieks starp tīmekļa serveriem, un blokķēdi. Šāda tehnoloģija teorētiski varētu tikt izmantota, piemēram, lai savāktu kvalitatīvus datus, vai pārbaudītu jau esošu datu precizitāti. Teorētiski šāda pieeja varētu palīdzēt atrisināt datu pilnīguma, precizitātes un datu nepretrunīguma problēmu.



3.9.att: **Shēma datu nodrošināšanai blokķēdē**

3.4. Kādas Lielo datu problēmas var atrisināt blokķēde?

Milzīgais Lielo datu apjoms, kā arī esošo risinājumu centralizācija atstāj atklātu jautājumu par tādu problēmu risināšanu kā drošība, privātums, kvalitāte, analīze un citas Lielo datu problēmas. Blokķēde apvieno tādas tehnoloģijas kā nemaināma virsgrāmata, šifrēšana, decentralizācija un viedie līgumi. Šajā nodaļā ir apskatīts, kā un kādas Lielo datu problēmas blokķēdes tehnoloģija teorētiski varētu atrisināt.

3.4.1. Lielo datu drošība un privātums

Vienots atteices punkts

Blokķēdes darbību nodrošina mezgli, tie pārbauda datus, transakcijas un blokus, kā arī pievieno jaunus blokus blokķēdē. Mezgli veido P2P tīklu, kas nodrošina decentralizāciju, tas nozīmē, ka nav viena servera, kas jebkurā brīdī var būt uzlauzts vai pārtraukt tā darbu un dati tajā brīdī nebūs pieejami.

DDoS uzbrukumi

Lielie dati vai, pareizāk sakot, serveri, kuros tie tiek glabāti, var tikt pakļauti DDoS uzbrukumiem. Blokķēdes tehnoloģija var mazināt šādus uzbrukumus un padarīt tos mazāk iespējamus, jo blokķēde ir decentralizēta, un visi mezgli koplieto vienu virsgrāmatu. Uzbrucējam ir grūti uzbrukt visus mezglus vienlaikus, lai sistēma neļautu sniegt pakalpojumus. Pat ja daži mezgli ir bojāti, visa sistēma var nepārtraukti sniegt pakalpojumus. Protams, tas problēmu pilnībā neatrisinās, jo mezgliem, kas nav pakļauti uzbrukumam, būs liela noslodze [26].

Jo lielāks ir mezglu skaits, jo grūtāk efektīvi implementēt DDoS uzbrukumu, uzbrukums kļūst finansiāli neizdevīgs, jo palielinās komisijas maksa.

DDoS mazināšana var novērst situācijas, kad dati nav nepieejami kādu laiku. Tas var būt svarīgi, piemēram, veselības aprūpes nozarē, kad ārsts nevar piekļūt pacientu datiem.

Datu audits

Pateicoties jaucējfunkcijām un kriptogrāfijai, izmaiņas blokķēdē kļūst gandrīz neiespējamas (joprojām pastāv 51% uzbrukuma iespējamība, bet tas ir atkarīgs no konsensa algoritma).

Tādējādi ikviens var redzēt transakcijas, kas notikuši blokķēdes vēsturē, un ikviens var laicīgi atgriezties pie 10 vai 20 gadiem un pārbaudīt, kas tur noticis.

Šis blokķēdes īpašība var atrisināt daudzas problēmas, kas ir saistītas ar uzticēšanos datu avotam. Piemēram, žurnālista vai zinātnieka publicēts darbs netiks viltots, jebkurš var pierādīt dokumenta izcelsmi, šis īpašums atrisina arī problēmu ar plaģiātu pateicoties laika zīmogam – var pierādīt, kurš pirmais publicēja darbu.

Datu nolaupīšana un noplūde

Ja dati tiek glabāti centralizēti, tad teorētiski jebkurš organizācijā var nozagt citu cilvēku privātos datus. Pat ja dati ir šifrēti, šai personai var būt piekļuve privātajai atslēgai. Blokķēdē katram datu sniedzējam ir sava privātā atslēga, ar kuru viņš šifrē savus datus.

Lielie dati nenodrošina personu privātumu

Blokķēdes tehnoloģija ļauj atbrīvoties no trešās puses līdzdalības, jo blokķēdi kontrolē sabiedrība, ikviens mezgls (atkarīgs no blokķēdes veida) var piedalīties blokķēdes struktūrā (balsot, apstiprināt transakcijas), to nodrošina konsenss mehānisms.

Kā arī neviens nezina, kur atrodas lietotāja dati, jo dati tiek šifrēti, pat ja sistēma tiks apdraudēta, dati nebūs pieejami. Datus var atšifrēt tikai ar privāto atslēgu.

Tā kā dati tagad pieder lietotājam, viņam ir iespēja pārvaldīt savus datus un izlemt, kuru datu daļu viņš grib koplietot, bez šaubām, ka viņa privātie dati, piemēram, adrese kļūs publiski pieejami. Tas var ietekmēt analīzes kvalitāti, jo analītiķiem kļūst pieejami vairāk datu. Uzraudzības iestādes (trešās puses) trūkums atrisina arī cenzūras problēmu, ikviens dalībnieks var sniegt jebkādu informāciju (ja vien viedajā līgumā nav norādīts citādi) [24].

3.4.2. Lielo datu kvalitātes kontrole

Kā minēts nodaļā “1.5.1. Datu apjoms”, pieaugot datu apjomam, uzņēmumiem kļūst grūti kontrolēt savākto un ģenerēto datu kvalitāti, gadījumā kad datu apjoms pieaug, organizācijām būs jāpalielina arī datu kvalitātes pārvaldības pasākumus. Šādas darbības var būt dārgas un grūti īstenojamas.

Turklāt, tā kā dati nāk no vairākiem avotiem, datu kvalitāte bieži atšķiras, tas arī apgrūtina datu kvalitātes kontroli organizācijām, jo īpaši tad, ja dati ir neapstrādāti, ir nepilnīgi vai satur daudz kļūdu.

Datu caurspīdīgums, integritāte, uzticamība un pilnīgums

Virsgrāmatas nemainīguma īpašība neļauj lietotājiem mainīt datus, kad tie ir pievienoti blokķēdei. Pastāv iespēja, ka cilvēks mainīs Lielo datu ierakstus, lai ietekmētu lielo datu analītikas prognozes [6], blokķēdē to ir gandrīz neiespējami izdarīt.

Kā arī pateicoties virsgrāmatas nemainīgumam un laikzīmogošanai, informācija par visām notikušām transakcijām, izveidotajiem blokiem un viņu izveidošanas datums un laiks glabājas blokķēdē. Lietotājs jebkurā brīdī var pierādīt, ka dati tika pievienoti noteiktā datumā un laikā.

Atsevišķos gadījumos viedais līgums var pārbaudīt, vai datu kvalitāte atbilst nepieciešamajiem kritērijiem un neļauj publicēt nepilnīgus, nekvalitatīvus datus.

Datu precizitāte

Decentralizēti orākuli var palīdzēt datu vākšanā. DON darbojas tā, ka ņem datus no vairākiem avotiem un atrod visprecīzāko informāciju. Piemēram, DON var ņemt laikapstākļu datus konkrētai pilsētai no vairākiem avotiem un atrast visbiežāk sastopamo vērtību, tādējādi nodrošinot datu precizitāti.

3.4.3. Lielo datu analīze

Mākslīgā intelekta centralizācija un sliktā datu kvalitāte

Mākslīgā intelekta metodes balstās uz savāktajiem datiem, un tās process var būt neefektīvs datu sliktas kvalitātes dēļ. Tā kā blokķēdē var uzlabot datu kvalitāti (sk. sadaļu “3.4.2 Lielo datu kvalitātes kontrole”), iegūtie mākslīgā intelekta analīzes rezultāti daudz ticami [6].

Viedie līgumi nodrošina iepriekš noteiktu nosacījumu izpildi. Ja apvienosim mākslīgo intelektu un viedos līgumus, tad mākslīgā intelekta aprēķinu rezultāts būs neapstrīdams un uzticams.

3.4.4. Kopsavilkums

3.2. tabulā tiek apkopotas visas minētās problēmas.

Kategorija	**Problēma**	**Iemesls**
Drošība un privātums	Datu nolaupīšana un noplūde	Dati tiek šifrēti un var tikt atšifrēti tikai ar privātu atslēgu. Konsenss mehānismi diezgan droši, lauzis var, piemēram, mēģināt veikt 51% uzbrukumu (atkarīgs no konsensa mehānisma)
	Datu audits	Virsgrāmata nevar tikt mainīta un glabā visas notikušas transakcijas, to nodrošina jaucējfunkcijas un konsensa mehānisms
	Vienots atteices punkts (angliski single point of failure)	Blokķēde ir decentralizēta, virsgrāmata tiek glabāta katrā pilnajā mezglā
	DDoS uzbrūkums	Blokķēde neaizsarga pilnīgi pret DDoS uzbrūkumu, bet var minimizēt to mīkstināt
	Lietotāja dati viņam nepieder	Konsenss mehānisms, lietotāja dati tiek šifrēti
Kvalitāte	Datu caurspīdīgums, integritāte, uzticamība un pilnīgums	Virsgrāmata nevar tikt mainīta, pateicoties laikzīmogošanai, lietotājs var pierādīt datu pievienošanas datumu un laiku, viedais līgums pārbauda datu kvalitāti
	Datu precizitāte	Orākuli var salīdzināt vairākus datu avotus un noteikt visprecīzāko informāciju.
Analīze	Mākslīgā intelekta algoritmu efektivitāte	Datu kvalitātes uzlabošana
	Mākslīgā intelekta centralizācija	Viedie līgumi nodrošina nosacījumu izpildi
Pārējās	Cilvēku iesaistīšana procesā	Iebūvēta ekonomikas sistēma

3.2.tabula: **Lielo datu problēmas, ko risina blokķēdes tehnoloģija**

3.5. Ierobežojumi

Blokķēdes tehnoloģija nebūt nav perfekta un, tāpat kā jebkura cita tehnoloģija, to implementēšanai ir arī trūkumi.

Piemēram, blokķēdes nemainīgums nodrošina datu uzticamību, bet arī rada datu uzglabāšanas problēmu. Tā kā Lieliem dati ir liels ātrums un līdz ar to liels transakciju skaits sekundē, palielināsies arī virsgrāmatas apjoms.

Lai IFPS sistēma varētu glabāt ZB informācijas, nepieciešams liels dalībnieku skaits un/vai ierīcēm, kurās darbojas mezgli, ir jābūt lielam RAM un NVME/SSD daudzumam.

Lietotāja privātā atslēga ir unikāla, un viņš pats ir atbildīgs par tās drošību. Ja lietotājs pazaudē privāto atslēgu, viņš zaudēs piekļuvi saviem datiem.

Blokķēdes risinājuma izstrāde ir diezgan dārga, pirms publiskās versijas palaišanas ir nepieciešams vairākas reizes pārbaudīt tīklu.

Viedie līgumi nevar pilnībā aprakstīt dažas sarežģītas problēmas cilvēku dzīvē. Reālajā dzīvē mēs saskaramies ar procesiem, kurus nevar pilnībā aprakstīt ar datoru, ar procesiem, kas prasa cilvēka mijiedarbību.

REZULTĀTI UN DISKUSIJA

Pētījuma laikā tika analizēti esošie Lielo datu pārvaldības risinājumi. Tādi risinājumi kā Apache Hadoop pārvalda datus centralizēti, kas rada noteiktus riskus: viena mezgla uzlaušana, veiksmīgu DDoS uzbrukumu iespējamība, datu zādzība no organizācijas pārstāvju puses, kā arī to ļaunprātīga izmantošana, u.c.

Līdz ar trešās paaudzes blokķēdes tehnoloģijas parādīšanos kļuva iespējams apstrādāt milzīgu (eksabaiti, petabaiti) datu apjomu. Jauni vienprātības mehānismi un starpķēžu platformas ir palielinājušas blokķēdes mērogojamību, caurlaidspēju, kā arī ļāvušas blokķēdei droši sazināties ar citām sistēmām.

Mezgla ierobežotā krātuves ietilpība neļauj glabāt Lielus datus blokķēdes virsgrāmatā, taču mūsdienu decentralizētās uzglabāšanas risinājumi ir balstīti uz dažādām izplātītām failu sistēmu struktūrām, kas tiek pārņemti no HDFS, IPFS, BitTorrent.

Pieaugot datu apjomam, uzņēmumiem ir grūti nodrošināt datu kvalitāti, blokķēdes tehnoloģija spēj uzlabot Lielot datu kvalitāti.

Pētījuma rezultātā tika piedāvātas dažas idejas decentralizētu Lielu datu problēmu vienkāršotam risinājumam, piemēram, Filecoin un Chainlink orākula izmantošana decentralizētai datu glabāšanas sistēmas nodrošināšanai un orākula izmantošana kvalitātes datu ieguvei.

SECINĀJUMS

Kopumā blokķēdes tehnoloģijām ir milzīgs potenciāls, lai palīdzētu atrisināt vairākas aktuālas Lielo datu problēmas, kas saistītas ar drošību, analīzi un apjomu. Tomēr, lai efektīvi atrisinātu šo problēmu, ir nepieciešams liels iesaistīto dalībnieku skaits. Arī jāsaprot, ka šāds risinājums var būt diezgan dārgs.

Darba procesā autors pilnveidoja zināšanas blokķēdes un Lielo datu jomā, kā arī pilnveidoja pētnieciskās prasmes.

IZMANTOTĀ LITERATŪRA UN AVOTI

[1] *How providing storage works*. Pieejams [tiešsaiste 30.05.2022]: https://docs.filecoin.io/storage-provider/how-providing-works/

[2] Chainlink, (14.09.2021). *What Is a Blockchain Oracle?* Pieejams [tiešsaiste 21.05.2022]: https://chain.link/education/blockchain-oracles

[4] Forbes, Tom Coughlin, *175 Zettabytes By 2025* (27.11.2018). Pieejams [tiešsaiste 22.05.2022]: https://www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025/?sh=36f9ce125459

[5] Statista Research Department, *Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025* (18.03.2022). Pieejams [tiešsaiste 22.05.2022]: https://www.statista.com/statistics/871513/worldwide-data-created/

[6] Deepa N, Quoc-Viet Pham, Dinh C. Nguyen, Sweta Bhattacharya, B. Prabadevi, Thippa Reddy Gadekallu, Praveen Kumar Reddy Maddikunta, Fang Fang, Pubudu N. Pathirana (5.02.2021). *A Survey on Blockchain for Big Data: Approaches, Opportunities, and Future Directions*,

[7] Ghotkar, M., & Rokde, P. (2016). *Big Data: How it is Generated and its Importance.* IOSR Journal of Computer Engineering.

[8] M. Hajirahimova, A. Aliyeva (2017). *About Big Data Measurement Methodologies and Indicators*. I.J. Modern Education and Computer Science. Pieejams [tiešsaiste 22.05.2022]: https://www.researchgate.net/publication/322066117_About_Big_Data_Measurement_Methodologies_and_Indicators

[9] Filecoin homepage. Pieejams [tiešsaiste 30.05.2022]: https://largedata.filecoin.io/

[10] A.Ņikiforova (2020). *Datu kvalitātes definēšana un novērtēšana*. Promocijas darbs, Latvijas Universitāte

[11] L.Cai, Y.Zhu (22.05.2015). *The Challenges of Data Quality and Data Quality Assessment in the Big Data Era*. Pieejams [tiešsaiste 23.05.2022]: https://datascience.codata.org/articles/10.5334/dsj-2015-002/

[12] R.Wang, M. Strong (1996). *Beyond Accuracy: What Data Quality Means to Data Consumers*. Journal of Management Information Systems. Pieejams [tiešsaiste 23.05.2022]: http://mitiq.mit.edu/Documents/Publications/TDQMpub/14_Beyond_Accuracy.pdf

[13] D.Becker, T.D.King, B.McMullen (2015), MITRE. *Big data, big data quality problem.* IEEE International Conference on Big Data.

[14] Satoshi Nakamoto (2008) *Bitcoin: A Peer-to-Peer Electronic Cash System*. Pieejams [tiešsaiste 23.05.2022]: https://bitcoin.org/bitcoin.pdf

[15] Margarita Parhomenko (2022). *Lietu Interneta un blokķēžu tehnoloģijas pielietošana piegāžu ķēžu pārvaldībā*. Kursa darbs, Latvijas Universitāte.

[16] Cynthia Dwork, Noni Naor (1993). *Pricing via Processing, Or, Combatting Junk Mail, Advances in Cryptology*. CRYPTO'92: Lecture Notes in Computer Science No. 740. Springer: 139–147. Pieejams [tiešsaiste 23.05.2022]: https://link.springer.com/content/pdf/10.1007/3-540-48071-4_10.pdf

[17] Ethereum mājaslapa. Pieejams [tiešsaiste 23.05.2022]: https://ethereum.org/en/

[19] Cardano projekta mājaslapa. Pieejams [tiešsaiste 23.05.2022]: https://cardano.org/discover-cardano

[20] Raynor de Best (24.03.2022). *Average transaction speed of 66 cryptocurrencies with the highest market cap as of March 2022*. Statista. Pieejams [tiešsaiste 24.05.2022]: https://www.statista.com/statistics/944355/cryptocurrency-transaction-speed/

[21] S.Singh, V.Vadi (2022). *Evolutionary Transformation of Blockchain Technology*. Pieejams [tiešsaiste 24.05.2022]: https://www.researchgate.net/publication/357766583_Evolutionary_Transformation_of_Blockchain_Technology

[22] Vidējais transakciju skaits blokā. Pieejams [tiešsaiste 24.05.2022]: https://www.blockchain.com/charts/n-transactions-per-block

[23] Storj. *Updates to Farmer Payouts and Network Testing* (2018). https://www.storj.io/blog/updates-to-farmer-payouts-and-network-testing

[24] Deepa, Pham, Nguyen, Bhattacharya, Prabadevi, Gadekallu, Reddy Maddikunta, Fang, Pathirana (2021). *A Survey on Blockchain for Big Data: Approaches, Opportunities, and Future Directions*. Pieejams [tiešsaiste 24.05.2022]: https://arxiv.org/pdf/2009.00858.pdf

[25] Cheng, Xu, Tang, Sheng, Cai (2018). *An Abnormal Network Flow Feature Sequence Prediction Approach for DDoS Attacks Detection in Big Data Environment.*

[26] Shah, Ullah, Li, Levula, Khurshid (2022). *Blockchain Based Solutions to Mitigate Distributed Denial of Service (DDoS) Attacks in the Internet of Things (IoT): A Survey*.

[27] J.Bicevskis, A.Nikiforova, Z.Bicevska, I.Oditis, G.Karnitis (2019). *A Step Towards a Data Quality Theory*. Faculty of Computing,University of Latvia, DIVI Grupa Ltd.

[28] L.Clissa (2022). *Survey of Big Data sizes in 2021*.

[29] Facebook privacy policy. Pieejams [tiešsaiste 26.05.2022]: https://www.facebook.com/about/privacy/previous

[30] Facebook cookie policy. https://chain.link/whitepaper https://www.facebook.com/policies/cookies/

[31] L.Breidenbach, C.Cachin, B.Chan, A.Coventry, S.Ellis, A.Juels, F.Koushanfar, A.Miller, B.Magauran, D.Moroz, S.Nazarov, A.Topliceanu1, F.Tram`er, F.Zhang (15.04.2021). *Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks*. Pieejams [tiešsaiste 26.05.2022]: https://chain.link/whitepaper

[33] S.Ellis, A.Juels, S.Nazarov (4.09.2017). *ChainLink A Decentralized Oracle Network*. Pieejams [tiešsaiste 26.05.2022]: https://research.chain.link/whitepaper-v1.pdf

[34] Statista (4.04.2022). *Bitcoin (BTC) blockchain size*. Pieejams [tiešsaiste 27.05.2022]: https://www.statista.com/statistics/647523/worldwide-bitcoin-blockchain-size/

[35] etherscan.io. *Ethereum full node sync chart*. Pieejams [tiešsaiste 27.05.2022]: https://etherscan.io/chartsync/chaindefault

[36] Ethereum (28.02.2022). *Decentralized strorage*. Pieejams [tiešsaiste 27.05.2022]: https://ethereum.org/en/developers/docs/storage/

[37] Vitalik Buterin (23.05.2021). *The Limits to Blockchain Scalability*. Pieejams [tiešsaiste 27.05.2022]: https://vitalik.ca/general/2021/05/23/scaling.html

[38] 15.3TB SSD drive on Amazon. Pieejams [tiešsaiste 27.05.2022]: https://www.amazon.com/TEAMGROUP-Internal-Compatible-Desktop-T253X7153T0C101/dp/B08SQLNHC8

[39] The DxChain Team. DxChain whitepaper (20.06.2018). *A Decentralised Big Data and Machine Learning Network Powered by a Computing-Centric Blockchain.* Pieejams [tiešsaiste 28.05.2022]: https://static.dxchain.com/web/DxChain-Whitepaper.pdf

[40] HDFS Architecture Guide (atjaunināts 2020). Pieejams [tiešsaiste 28.05.2022]: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[41] Hadoop MapReduce tutorial (atjaunināts 2020). Pieejams [tiešsaiste 28.05.2022]: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

[42] Hadoop YARN (atjaunināts 2022). Pieejams [tiešsaiste 28.05.2022]: https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html

[43] Juan Benet (2014). *IPFS - Content Addressed, Versioned, P2P File System*. Pieejams [tiešsaiste 28.05.2022]: https://arxiv.org/pdf/1407.3561.pdf

[44] Protocol Labs (19.07.2017). *Filecoin: A Decentralized Storage Network*. Pieejams [tiešsaiste 28.05.2022]: https://filecoin.io/filecoin.pdf

[45] Storj Labs, Inc (30.10.2018). *Storj: A Decentralized Cloud Storage Network Framework*. https://www.storj.io/storjv3.pdf

[46] Statista (2016). *What are the main causes for poor data quality?*. Pieejams [tiešsaiste 28.05.2022]: https://www.statista.com/statistics/518069/north-america-survey-enterprise-poor-data-quality-reasons/

[47] Bitcoin fees per transaction. Pieejams [tiešsaiste 28.05.2022]: https://www.blockchain.com/charts/fees-usd-per-transaction

[48] Ethereum fees per transaction. Pieejams [tiešsaiste 28.05.2022]: https://ycharts.com/indicators/ethereum_average_transaction_fee

[49] Cardano fees per transaction. https://messari.io/asset/cardano/chart/txn-fee-avg