ISSN 2224-087X. Електроніка та інформаційні технології. 2024. Випуск 28. С. 73–83
Electronics and information technologies. 2024. Issue 28. P. 73–83

УДК 004.6 DOI: https://doi.org/10.30970/eli.28.7

INTEGRATING SOCIAL MEDIA DATA INTO SMART CONTRACTS

Yurii Tsudzenko

Ivan Franko National University of Lviv,
50 Drahomanova St., UA-79005 Lviv, Ukraine
Yurii.Tsudzenko@lnu.edu.ua

The rise of blockchain technology has set the stage for groundbreaking decentralized applications and smart contracts. Recently, there's been a surge in interest regarding the integration of social media data into blockchain-based smart contracts, promising significant transformations across sectors like finance, marketing, and governance. Essentially, this shift involves tapping into the vast pool of data generated by social media interactions immutable smart contracts. By capitalizing on blockchain transparency, security, and decentralization, this integration aims to streamline processes, foster trust, and unlock new avenues for automation and efficiency.

This paper delves into the process of gathering data from YouTube, a prominent video-sharing platform, via its API for use in an integrating data into smart contract. YouTube boasts an extensive repository of data ripe for various applications. The research utilizes Node.js, Solidity, and YouTube API technologies. Furthermore, it explores incorporating this gathered information into a smart contract, enriching features within a decentralized ecosystem. The incorporation of social media data into a smart contract offers fresh prospects for data-driven decision-making and content verification, contributing to the advancement of blockchain-based applications and services. The process of converting data into smart contracts is divided into several main stages. The article also provides the results of execution time testing for transferring data from social media into smart contracts. The conducted tests showed a significant reduction in execution time thanks to the utilization of the YouTube API along with Node.js and Solidity technologies. This approach to integrating data into smart contracts can be applied for further analysis and content verification, fostering the development of blockchain-based applications and services.

Keywords: cryptocurrency, smart contract, Solidity, social media, decentralization, data analysis.

Introduction.

Smart contracts based on blockchain technology open up new opportunities for secure and automated data exchange. This technology has been widely used in many fields, such as banking, social media and mand in many another areas. Transferring data into smart contracts is becoming the backbone of the security and efficiency of digital transactions, but this issue is not always simple. Integrating smart contracts with external data sources is an important aspect of blockchain technology, as it enables decentralized applications to interact with real-world data.

Overall, addressing the challenge of efficient and secure data transfer to smart contracts requires careful consideration of data integrity, format, transmission methods, and gas costs [1, 2]. By implementing robust security measures and using appropriate technologies, it is possible to ensure the efficient interaction of smart contracts with external data sources, while maintaining the integrity and security of the blockchain network [3]. This can include text strings, numeric values, structured objects such as JSON, or even references to external data.

Related work.

In this section the potential widespread applications of smart contracts across various domains are explored. Previous studies have extensively discussed their versatility and implications. Additionally, there is a notable discussion regarding the tools available for crafting smart contracts, alongside an examination of emerging trends in Bitcoin [4, 5]. One significant aspect highlighted in this paper is the utilization of data signature methods for securely retrieving data, relying on QR code generation as part of the signature process [6].

However, incorporating data retrieved directly from the YouTube API into a blockchain-based smart contract poses certain challenges. Despite the considerable attention smart contracts have garnered from researchers, the volume of published literature remains relatively limited. This article delves into a method for integrating real-time data from social media into smart contracts, a job that presents its own set of complexities and metrics.

Research methodology.

YouTube was selected from the existing social platforms for data mining. YouTube stands as more than just a video-sharing platform; it's a versatile social network where users can both upload their own content and engage with others in real-time [7]. Often labeled as a social media platform, YouTube's interactive and community-driven essence sets it apart. While its core function revolves around sharing videos, it incorporates numerous features synonymous with social media [8]. Through tools like comments, likes, dislikes, shares, and subscriptions, YouTube fosters user engagement and interaction. Individuals can connect with content creators and fellow viewers by leaving feedback, reacting to videos, and circulating content across their networks. This is why YouTube is considered a social media platform [9, 10].

In order to get data from YouTube, the YouTube API [11] is used. This API provides a set of tools and protocols that enable developers to programmatically access and interact with YouTube's features and data. This data may encompass various elements such as user profiles, videos, playlists, comments, likes, and other relevant information.

Fig.1. Process of data transformation from social media to smart contract

Implementation of data transfer into a smart contract involves dividing the transformation process into several stages [14]. Collecting data from social media for use in a smart contract involves several steps and stages due to the decentralized and potentially sensitive nature of the data. However, we can represent the transformation process conceptually. Let's break down the pie chart into different stages of the transformation process. Each of this process take part of data integration:

1. *Data Collection* stage involves gathering various types of data from different sources such as databases, APIs, user input, etc., accounting for about 15% of the process.
2. *Data Processing* makes up 15% of the process, once collected, the data needs to be processed to ensure its accuracy, consistency, and relevance. This could involve cleaning the data, removing duplicates, handling missing values, etc.
3. *Data Transformation* stage, constituting 25% of the process, is where the actual transformation into smart contracts occurs. It involves converting the processed data into a format suitable for smart contract execution. This may include defining contract structures, encoding data, and implementing security measures.
4. *Smart Contract Deployment*: after the transformation, smart contracts are deployed onto the blockchain network, making up 20% of the process. This stage involves interacting with blockchain platforms, deploying the contracts, and configuring necessary parameters.
5. *Execution and Monitoring* comprising 20% of the process, once deployed, smart contracts execute predefined functions autonomously. Monitoring is crucial to ensure that contracts behave as expected, handle transactions correctly, and adhere to predefined rules.
6. *Maintenance and Updates*: smart contracts may require maintenance and updates over time to adapt to changing requirements, fix bugs, or implement new features. This stage, accounting for 5% of the process, involves managing contract versions, upgrading contracts, and handling backward compatibility.

Figure 2 depicts all the stages involved in extracting data from YouTube and integrating it into a smart contract, as listed above, in sequence. Additionally, each of these stages conducts data transfer and ensures trust and transparency within the blockchain ecosystem.

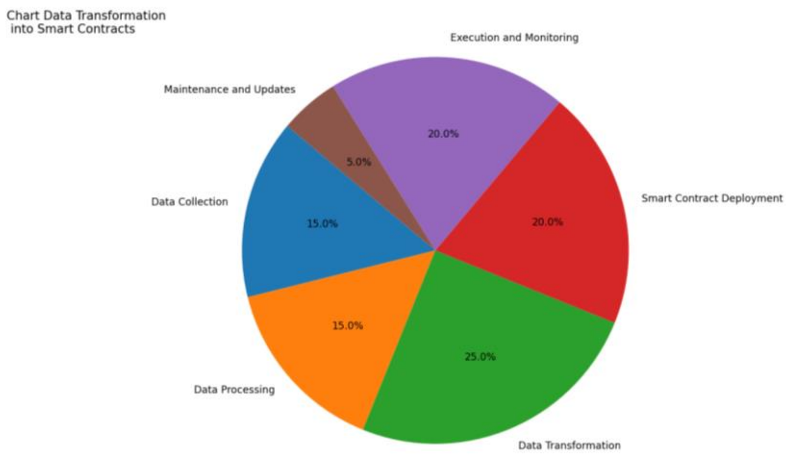

Fig. 2. Chart of process of data exchange into smart contract.

Results and discussion.

Data extraction from the YouTube API can be implemented using various programming languages [11]. For this research, Node.js [12] was chosen to facilitate API communication and

ISSN 2224-087X. Електроніка та інформаційні технології. 2024. Випуск 28

data retrieval. Figures 2 and 3 showcase the results obtained from querying the YouTube API [11].


```
[{"kind":"youtube#searchResult","etag":"DAdRjVIWmJcWlLjAAJSAHvUvi-Y","id":{"kind":"youtube#channel","channelId":"U
21T07:36:31Z","channelId":"UCJWh7F3AFyQ_x01VKzr9eyA","title":"Smart Contract Programmer","description":"This chann
our adventures where ...","thumbnails":{"default":{"url":"https://yt3.ggpht.com/PUHc3k1kptoEzGifgj8PmX2kkJS6GLVehH
{"url":"https://yt3.ggpht.com/PUHc3k1kptoEzGifgj8PmX2kkJS6GLVehH3EY8Qdhz8clI1jgb4ERMeoQe-I2JfSIFCAxRCe=s240-c-k-c0
{"url":"https://yt3.ggpht.com/PUHc3k1kptoEzGifgj8PmX2kkJS6GLVehH3EY8Qdhz8clI1jgb4ERMeoQe-I2JfSIFCAxRCe=s800-c-k-c0
Programmer","liveBroadcastContent":"none","publishTime":"2019-04-21T07:36:31Z"}},{"kind":"youtube#searchResult","e
{"kind":"youtube#video","videoId":"M576WGiDBdQ"},"snippet":{"publishedAt":"2021-09-09T13:38:22Z","channelId":"UC8b
Expert Python Tutorial","description":"This course will give you a full introduction into all of the core concepts
{"url":"https://i.ytimg.com/vi/M576WGiDBdQ/default.jpg","width":120,"height":90},"medium":{"url":"https://i.ytimg.
{"url":"https://i.ytimg.com/vi/M576WGiDBdQ/hqdefault.jpg","width":480,"height":360}},"channelTitle":"freeCodeCamp.
{"kind":"youtube#searchResult","etag":"fJy-G0NKyNv8M18hjo9UegJsAxA","id":{"kind":"youtube#playlist","playlistId":"
14T11:27:41Z","channelId":"UCJWh7F3AFyQ_x01VKzr9eyA","title":"Solidity 0.8","description":"","thumbnails":{"defaul
{"url":"https://i.ytimg.com/vi/xv9OmztShIw/mqdefault.jpg","width":320,"height":180},"high":{"url":"https://i.ytimg
Programmer","liveBroadcastContent":"none","publishTime":"2021-09-14T11:27:41Z"}},{"kind":"youtube#searchResult","e
{"kind":"youtube#video","videoId":"umepbfKp5rI"},"snippet":{"publishedAt":"2023-06-07T13:14:12Z","channelId":"UCn-
Powered By AI - Full Course (0 - 6)","description":"This course will give you a full introduction into all of the
{"default":{"url":"https://i.ytimg.com/vi/umepbfKp5rI/default.jpg","width":120,"height":90},"medium":{"url":"https
{"url":"https://i.ytimg.com/vi/umepbfKp5rI/hqdefault.jpg","width":480,"height":360}},"channelTitle":"Patrick Colli
{"kind":"youtube#searchResult","etag":"F9iKaagBcNGckVftAwxRUINbxvU","id":{"kind":"youtube#video","videoId":"ZE2HxT
20T20:00:03Z","channelId":"UCnxrdFPXJMeHru_b4Q_vTPQ","title":"Smart contracts - Simply Explained","description":"W
let's find in plain English!","thumbnails":{"default":{"url":"https://i.ytimg.com/vi/ZE2HxTmxfrI/default.jpg","wid
{"url":"https://i.ytimg.com/vi/ZE2HxTmxfrI/mqdefault.jpg","width":320,"height":180},"high":{"url":"https://i.ytimg
Explained","liveBroadcastContent":"none","publishTime":"2017-11-20T20:00:03Z"}},{"kind":"youtube#searchResult","et
{"kind":"youtube#video","videoId":"pyaIppMhuic"},"snippet":{"publishedAt":"2021-05-09T18:33:44Z","channelId":"UCsY
Animated)","description":"Are you wondering what a Smart Contract is? Well, a smart contract is an agreement betwe
{"url":"https://i.ytimg.com/vi/pyaIppMhuic/default.jpg","width":120,"height":90},"medium":{"url":"https://i.ytimg.
{"url":"https://i.ytimg.com/vi/pyaIppMhuic/hqdefault.jpg","width":480,"height":360}},"channelTitle":"Whiteboard Cr
{"kind":"youtube#searchResult","etag":"LbzNjXhStCEF71wKd0ugplPqV34","id":{"kind":"youtube#video","videoId":"RQzuQb
02T19:00:07Z","channelId":"UCY0xL8V6NzzFcwzHCgB8orQ","title":"Learn Solidity in 20 Minutes!","description":"Become
Intro 01:00 What ...","thumbnails":{"default":{"url":"https://i.ytimg.com/vi/RQzuQb0dfBM/default.jpg","width":120,
{"url":"https://i.ytimg.com/vi/RQzuQb0dfBM/mqdefault.jpg","width":320,"height":180},"high":{"url":"https://i.ytimg
University","liveBroadcastContent":"none","publishTime":"2022-09-02T19:00:07Z"}},{"kind":"youtube#searchResult","e
toX0"},"snippet":{"publishedAt":"2022-06-19T13:00:15Z","channelId":"UCn-3f8tw_E1jZvhuHatROwA","title":"How to Audi
learn the auditing 101 of smart contracts, and understand basic tooling for doing an audit, like slither, echidna,
```


Fig. 3. Raw data get from social media

Upon collection, the acquired data undergoes processing to extract essential information and is formatted in a manner suitable for integration into a smart contract. This data encompasses details relevant to the topic of smart contracts video queries, including video titles, thumbnail images, publication timestamps, descriptions, video links, unique identifiers, channel names, and associated comments. Figure 2 illustrates the stored video data, comprising approximately 1000 unique items, structured in JSON format [13].

On Figure 3, receiver data from another request regarding comments and videos is presented in a visually appealing format. Figure 4 showcases the code demonstrating how to establish a connection via Node.js to the YouTube API.

After extracting and analyzing data from social media, the subsequent stage involves utilizing a smart contract platform for data exchange. Solidity [15, 16] is chosen as the platform for transferring the data. For the research, 1000 unique elements were selected and transferred into a smart contract using Node.js and Solidity. All tests were conducted on a Windows operating system, utilizing a system with 16GB of RAM and an Intel Core i5 processor. The results of these tests are presented in Table 1.

```json
  {
    "kind": "youtube#searchResult",
    "etag": "Rt6oJq9C__woLEgMROXL6TkhuIw",
    "id": {
      "kind": "youtube#video",
      "videoId": "pyaIppMhuic"
    },
    "snippet": {
      "publishedAt": "2021-05-09T18:33:44Z",
      "channelId": "UCsYYksPHiGqXHPoHI-fm5sg",
      "title": "What are Smart Contracts in Crypto? (4 Examples + Animated)",
      "description": "Are you wondering what a Smart Contract is? Well, a smart contract
      "thumbnails": {
        "default": {
          "url": "https://i.ytimg.com/vi/pyaIppMhuic/default.jpg",
          "width": 120,
          "height": 90
        },
        "medium": {
          "url": "https://i.ytimg.com/vi/pyaIppMhuic/mqdefault.jpg",
          "width": 320,
          "height": 180
        },
        "high": {
          "url": "https://i.ytimg.com/vi/pyaIppMhuic/hqdefault.jpg",
          "width": 480,
          "height": 360
        }
      },
      "channelTitle": "Whiteboard Crypto",
      "liveBroadcastContent": "none",
      "publishTime": "2021-05-09T18:33:44Z"
    }
  },
  {
    "kind": "youtube#searchResult",
```


Fig. 4. Data Received from YouTube API.

```javascript
import express from "express";
import { google } from "googleapis";
import 'dotenv/config';

const app = express();
const port = process.env.port;
const apiKey = process.env.apiKey;
const baseApiUrl = "https://www.googleapis.com/youtube/v3";
const youtube = google.youtube({
  version: 'v3',
  auth: apiKey,
})

app.get("/search-with-googleapis", async (req, res, next) => {
  try {
    const searchQuery = req.query.search_query;

    const response = await youtube.search.list({
      part: 'snippet',
      q: searchQuery,
      type: 'video',
      maxResults: 100
    })
    const titles = response.data.items.map((item) => item.snippet.title);
    console.log(titles.length);
    res.send(response.data.items);
  } catch (err) {
    next(err);
  }
});

app.listen(port, () => {
  console.log(`Server start on port ${port}`);
  console.log()
});
```

Fig. 5. Method to Retrieve Data from YouTube API.

Table 1. Time extract items from API to smart contract.

Count of items	Time (s)
1	0.04
10	1.5
100	35.8
500	440.2
1000	672.8

In Fig. 6 the plot of dependence of execution time on the number of objects. For a huge amount of data integrate, the relationship between execution time and the number of objects retrieved might not be linear.

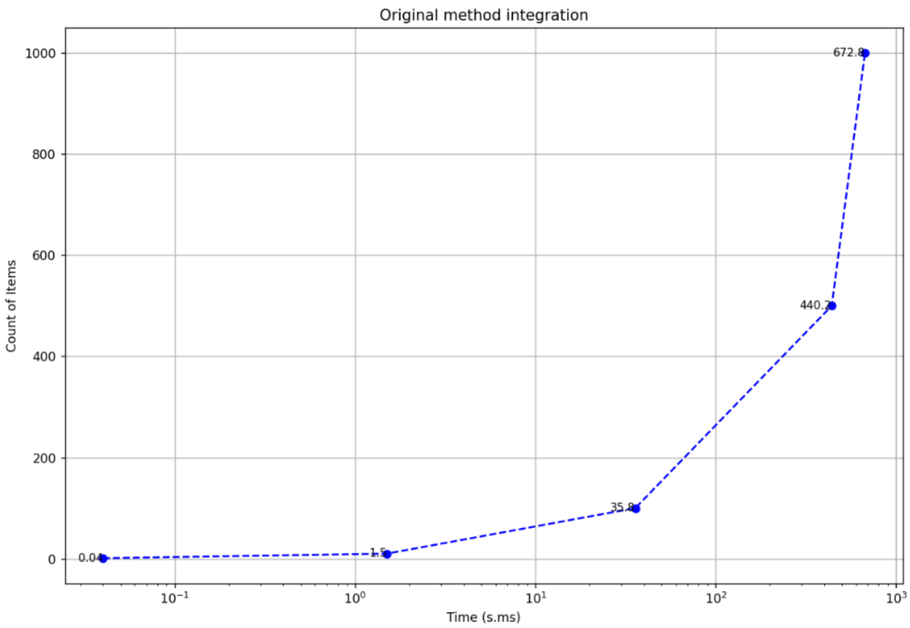

Fig. 6. Plot of extracted data from YouTube API into smart contract

The x-axis represents the time taken for extraction, measured in seconds, ranging from 0.04 to 672.8 seconds, while the y-axis represents the count of items extracted from the API, ranging from 1 to 1000. The line graph illustrates the relationship between the count of items and the time taken to extract them from the API to the smart contract. As the count of items increases, there is a noticeable trend of the time taken also increasing. Initially, with 1 item, the extraction time is minimal, at approximately 0.04 seconds. However, as the count of items escalates to 10, 100, 500, and 1000, the extraction time significantly increases, reaching 1.5 seconds, 35.8 seconds, 440.2 seconds, and 672.8 seconds, respectively. This suggests a non-linear relationship between the count of items and the extraction time, indicating that the extraction process experiences diminishing efficiency gains as the count of items grows. This relationship is visually depicted in Figure 6.

Table 2 presents a comparative analysis of execution times for various data integration methods, including original, caching, and compression techniques. The table showcases the efficiency gains achieved by caching and compression methods compared to the baseline execution times of the original method across different counts of items.

Table 2. Comparison of execution times for data integration methods

Count of Items	Original Time (s)	Compression Time (s)	Caching Time (s)
1	0.04	0.0133	0.008
10	1.5	0.7	0.58
100	35.8	12.93	9.16
500	440.2	166.73	120.04
1000	672.8	236.27	184.56

Compression reduces the size of the data being processed, which decreases the time required to transfer and store the data. By compressing data before processing, we can achieve faster integration times. The result is shown in Figure 7.

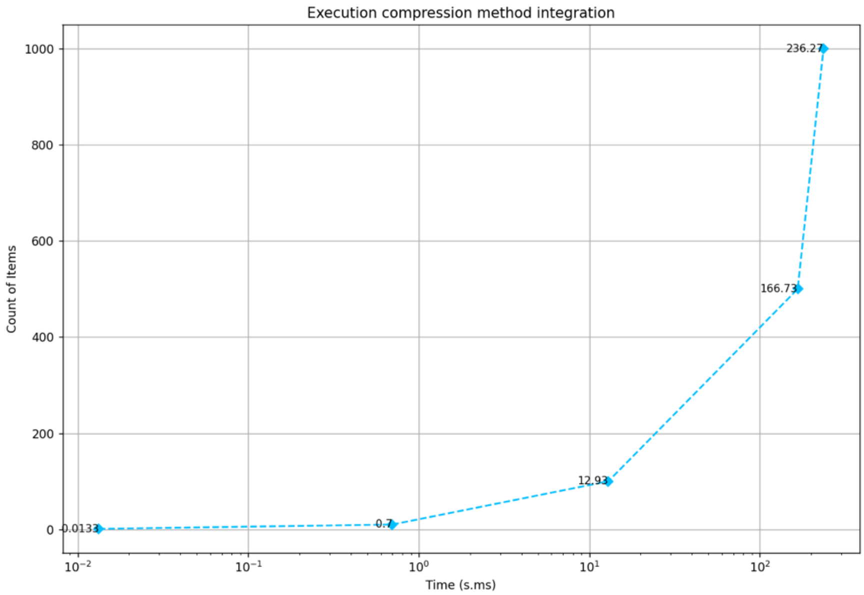

Fig. 7. Compression method for data integration

Caching involves storing frequently accessed data in a temporary storage area, which reduces the need to repeatedly fetch or compute the same data. The result is shown in Figure 8.

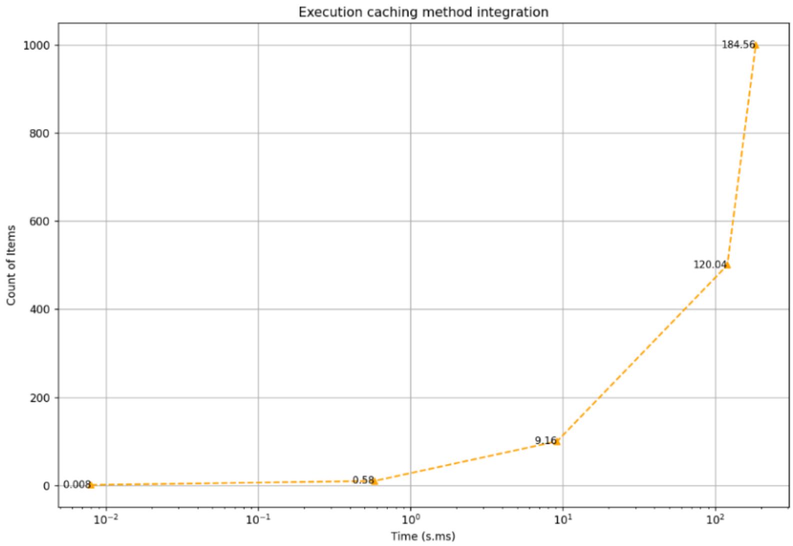

Fig. 8. Caching method of data integration

Incorporating caching and compression into the data integration process yields substantial performance gains. Caching provides immediate access to frequently used data, significantly reducing integration times. Compression minimizes data size, making the transfer and processing more efficient. These optimizations are crucial for handling large volumes of data in a timely and resource-efficient manner, making them essential techniques for modern data integration tasks.

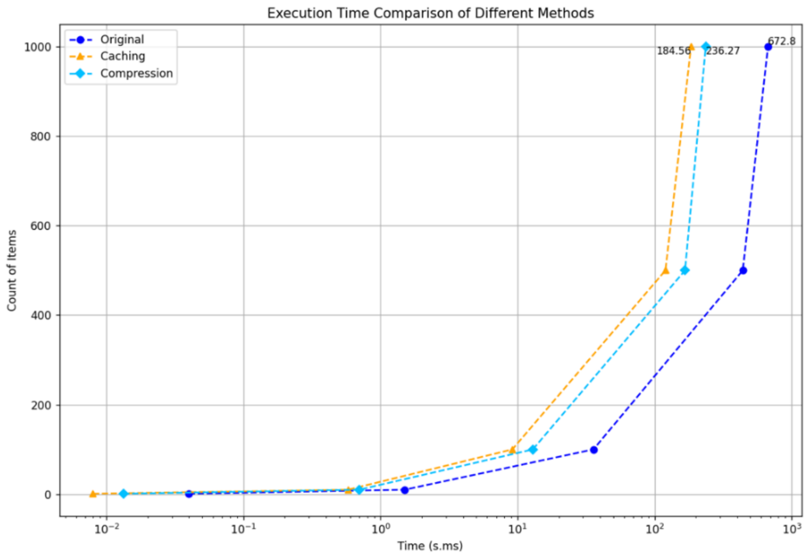

Fig. 9. Time efficiency of different integrating methods

Incorporating caching and compression into the data integration process yields substantial performance gains. Caching provides immediate access to frequently used data, significantly reducing integration times. Compression minimizes data size, making the transfer and processing more efficient. These optimizations are crucial for handling large volumes of data in a timely and resource-efficient manner.

This type of data processing through smart contracts offers several advantages. Data transmitted to smart contracts benefits from cryptographic security, ensuring protection against tampering and unauthorized alterations. This cryptographic security makes it practically impossible to modify the data without the appropriate permissions, thereby enhancing the integrity and trustworthiness of the data. Smart contracts can automatically perform actions based on input, simplifying the process of deals and transactions. The history of all transactions and data operations is recorded in the blockchain, which makes processes more transparent and inaccessible to falsification.

Conclusions.

This article examines the method of transferring real data extracted from social networks into smart contracts. During the research, data was obtained from social networks into smart contracts using YouTube API tools, Solidity based on the node.js framework. Smart contracts provide immutable storage, ensuring the integrity of data retrieved from JSON. Once data is stored on the blockchain, it cannot be altered or tampered with, enhancing trust in the retrieved information. Smart contracts can automate processes based on data retrieved from JSON, executing predefined actions when specific conditions are met. This automation can streamline operations and reduce the need for manual intervention. By storing JSON data on a blockchain, transparency is enhanced as all transactions and data modifications are recorded and visible to relevant parties. This transparency can improve accountability and trust in the system. Transferring data into smart contracts creates a secure and reliable environment for information exchange, which can facilitate the development of many digital applications in the future.

References

[1] *Alharby, Maher, and Aad Van Moorsel*. Blockchain-based smart contracts: A systematic mapping study. *arXiv preprint arXiv:1710.06372*. 2017, doi:10.48550/arXiv.1710.06372

[2] *A. Elvira, J. Correas, P. Gordillo, G. Román-Díez, A. Rubio*. Gasol: Gas analysis and optimization for ethereum smart contracts. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 118-125. Cham: Springer International Publishing, 2020, doi: 10.1007/978-3-030-45237-7

[3] *W. Wenbo, D. Thai Hoang, P. Hu, Zehui Xiong, D. Niyato, P. Wang, Y. Wen, D In Kim*. A survey on consensus mechanisms and mining strategy management in blockchain networks. *Ieee Access* 7 (2019): 22328-22370, doi**:** 10.1109/ACCESS.2019.2896108

[4] *Pavlyshenko B. M*. "Bitcoin price predictive modeling using expert correction." In *2019 XIth International Scientific and Practical Conference on Electronics and Information Technologies (ELIT)*, pp. 163-167. IEEE, 2019, **doi:** 10.1109/ELIT.2019.8892303

[5] *Yuriy Tsudzenko "*Підходи в моделюванні смарт-контрактів на основі Ethereum*" Electronics and information technologies*. 2023. Issue 22. P. 69–78.

[6] *Ali, Ammar Mohammed, Alaa Kadhim Farhan*. Enhancement of QR code capacity by encrypted lossless compression technology for verification of secure E-Document. *IEEE Access* 8 (2020): 27448-27458, doi: 10.1109/ACCESS.2020.2971779

[7] YouTube [Online]. URL: https://www.youtube.com/

[8] *Cheng X., Dale C., Liu J.* Statistics and social network of youtube videos. In *2008 16th Interntional Workshop on Quality of Service*. IEEE 2008, June. P. 229-238, https://ieeexplore.ieee.org/document/4539688

[9] *Chattopadhyay, Biswapesh, Priyam Dutta, Weiran Liu, Ott Tinn, Andrew Mccormick, Aniket Mokashi, Paul Harvey et al.* Procella: Unifying serving and analytical data at YouTube. *Proceedings of the VLDB Endowment* 12, no. 12, 2019: P. 2022-2034, doi: 10.14778/3352063.3352121

[10] *Bärtl, Mathias*. "YouTube channels, uploads and views: A statistical analysis of the past 10 years." *Convergence* 24, no. 1, 2018, P. 16-32. doi: https://doi.org/10.1177/1354856517736979

[11] Youtube Data API [Online]. URL: https://developers.google.com/youtube/v3

[12] Node.js [Online]. URL: https://nodejs.org/en

[13] *Pezoa, Felipe, Juan L. Reutter, Fernando Suarez, Martín Ugarte, Domagoj Vrgoč.* Foundations of JSON schema. In *Proceedings of the 25th international conference on World Wide Web*. P. 263-273. 2016, doi: 10.1145/2872427.2883029

[14] *Steelman, Zachary R., Bryan I. Hammer, and Moez Limayem*. "Data collection in the digital age." *MIS quarterly* 38, no. 2 (2014): 355-378.

[15] *Dannen Chris, Chris Dannen*. Solidity programming. *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*, 2017. P. 69-88, doi: https://10.1007/978-1-4842-2535-6

[16] *Wohrer Maximilian, Uwe Zdun*. "Smart contracts: security patterns in the ethereum ecosystem and solidity." In *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*. P. 2-8. IEEE, 2018, **doi:** 10.1109/IWBOSE.2018.8327565

ПЕРЕВЕДЕННЯ ДАНИХ ІЗ СОЦІАЛЬНИХ МЕРЕЖ У СМАРТ-КОНТРАКТИ

Юрій Цудзенко

Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, 79005 Львів, Україна
Yurii.Tsudzenko@lnu.edu.ua

Розвиток технології блокчейн створив основу для новаторських децентралізованих програм і смарт-контрактів. Останнім часом спостерігається зростаючий інтерес до інтеграції даних соціальних медіа в смарт-контракти на основі блокчейну, що передбачає значні перетворення в таких секторах, як фінанси, маркетинг і управління. Суттєвим аспектом цієї зміни є використання величезного обсягу даних, що генеруються в соціальних мережах, а також незмінність смарт-контрактів. За допомогою прозорості, безпеки та децентралізації блокчейну, ця інтеграція спрямована на спрощення процесів, підвищення рівня довіри та відкриття нових можливостей для автоматизації та підвищення ефективності.

У цій статті розглядається процес збору даних з відомої соціальної платформи YouTube для обміну відео, повідомлень, що містить велике сховище даних, через її прикладний програмний інтерфейс. Для дослідження використано технології Node.js, Solidity та YouTube API. Крім того, він досліджує можливість включення цієї зібраної інформації в смарт-контракт, збагачуючи функції в рамках децентралізованої екосистеми.

Інтеграція даних соціальних медіа в смарт-контракт відкриває нові перспективи для прийняття рішень на основі даних та перевірки вмісту, сприяючи розвитку додатків і послуг на основі блокчейну. У статті описано основні етапи процесу переведення даних у смарт-контракти. Процес переведення даних у смарт-контракти поділяється на декілька основних етапів, детально описаних у статті. Також у статті наведено результати тестування часу виконання для переведення даних із соціальних мереж у смарт-контракти. Проведені тести показали значне скорочення часу виконання завдяки використанню YouTube API разом із технологіями Node.js і Solidity. Такий підхід інтеграції даних у смарт-контракти може бути застосований для подальшого аналізу та перевірки вмісту, сприяючи розвитку додатків і послуг на основі блокчейну.

Ключові слова: криптовалюта, смарт-контракт, Solidity, соціальні медіа, децентралізація, аналіз даних.

The article was received by the editorial office on 18.06.2024.
Accepted for publication on 30.09.2024.