G-DaM: A Blockchain based Distributed Robust Framework for Ground Water Data Management

Sukrutha L. T. Vangipuram*, Saraju P. Mohanty*, Elias Kougianos†, and Chittaranjan Ray‡
* Department of Computer Science and Engineering, University of North Texas, USA
† Department of Electrical Engineering, University of North Texas, USA
‡ Nebraska Water Center, Water for Food Global Institute, University of Nebraska-Lincoln, USA
Email: lakshmisukruthatirumalavangipuram@my.unt.edu, saraju.mohanty@unt.edu,
elias.kougianos@unt.edu, cray@nebraska.edu

***Abstract*—In this paper, we examine the challenges of groundwater data flow management due to institutional barriers and constraints through technology. We load the groundwater statistics from end-systems towards Distributed Data Storage (DDS) and the blockchain (BC), implementing a dual hashing procedure through Infura gateways and smart contracts. The smart contracts, logical code, and functions give control over access and file sharing. With data secured through crypto puzzles, the quality and integrity are increased while simultaneously preventing higher fee charges for storage on BC with reduced cost and time.**

***Index terms*—** Smart Agriculture; Blockchain (BC); Distributed Data Storage (DDS); Groundwater data management.

I. Introduction

Groundwater is a source sustenance and living for many species on earth, including humans. It constitutes 1.69% of the total water present on earth's aquifers that are extracted using wells. Aquifers are recharged by rain and snowmelt [1]. Data is the key driving force for science. Groundwater data can be available from studies realated to aquifer property assessment, regional hydrogeologic characterization, and studies that focus on climate science, environmental science, public policy, and law. However, considering all these multiple contexts and diverse data sources is critical; integrating and combining all these data into a single intelligent framework is also a significant challenge. There are some new political issues raised for accessing groundwater data due to the limited engagement of stakeholders in sharing knowledge and technology. More data collection in agriculture enhances chances to increase food production with information on water availability. These facts allow researchers to perform simulation models and visualizations for predicting water supplies, developing groundwater reserves and preserving water levels for future generations. However, incorrect data result in inaccurate models that may lead to poor deductions. Analysts are mainly worried about the authenticity of the data, specifically because data sent through Web, from the districts to regional reasearch centers can be more subjected to tampering and modified quality through hackers at multiple places along the data path [2]. Validating data gathered can also be a significant problem when people have not paid close attention in collecting and digitally entering the data. The blockchain can be a potential answer for the experts to avoid uncertain data integrity and quality problems.

Central cloud data systems have a single-point failure, risk to data confidentiality, dependence on Internet connection, high latency, less security level, and vast opportunity for attacks on data [3]. In addition, data units having multiple formats require a system that can use one mode to store and share. Fig. 1 lists out some of the challenges included in the management of groundwater data. To overcome and address the above issues, using advanced technologies like the blockchain and distributed data storage could provide benefits.

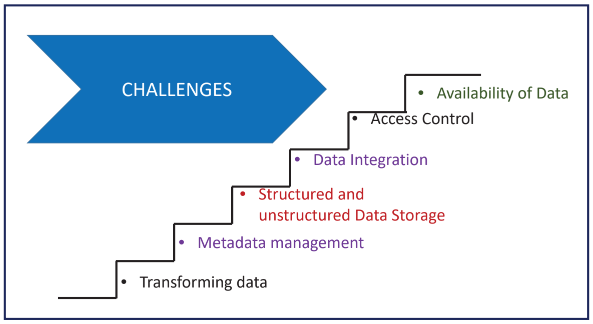


Fig. 1: Groundwater data management challenges.

.

The paper is organized as follows: We first address the problems and propose a solution with novelty in Section II. We then describe prior related works and different data sources in the groundwater sciences in Section III and Section IV, respectively. Section V discusses recipients of the groundwater and groundwater data, Section VI, gives a detailed groundwater data management system with unique solutions embedded. Algorithms for present data flow are explained in Section VII followed by implementation and experimental results for validating the current system in Section VIII, and Section IX, respectively Finally, Section X concludes the paper with a future research discussion.

II. Novel Contributions

- The blockchain is used as a unified and synchronized technology for groundwater data flow to mitigate uncertain facts and improve data quality.
- DDS for larger data distributed storage and added security for aquifer water level and water quality data.
- We propose a novel architecture for a water-quality data management system with double hashing refuge.
- Results comparing time, fee, and cost for traditional BC and BC with DDS are presented.

III. Prior Related Works

Various approaches have been explored and practiced for smart agriculture data to be integrated with immutability and privacy. In order to avoid market risk between consumers and producers, [4] proposes a new method that uses blockchain technology to verify and shield data. It can track the products coming to consumers from agricultural dairy farmers and imroves the data visibility and reliability. [5] investigates and stores the sensor data and events of farming in a blockchain and develops smart contracts with the help of Ethereum BC to buy the agricultural lands. The design provides quality of seeds, air and soil temperature data, sale price, soil moisture content, and trust and transparency. The system in [6] makes use of IoT and BC technologies to give payments and incentives to those farmers who use optimal qualtity of water for agriculture. The structure is designed to directly take the water usage data to BC networks without causing any loads on the IoT device.

IV. Sources for Groundwater Data

The primary information contains intricate fact content with vast amounts of data. They can relate to spatial, multimedia, remote sensing, and other data sources. Not all the data coming from different sources has a similar structure; for instance, GIS (Geo-information System) or geospatial data for nitrogen use [11] for crops differs from text(numerical) data collected; so the methods and tools used to provide storage and security for each of these data types are different. Fig. 2 shows the locations for groundwater monitoring sites of the United States, and Fig. 3 illustrates the usage of water in billions of gallons per day. The latest year data for groundwater usage is collected from the U.S Geological Survey [12] is for 2015. Before implementing solutions, a complete understanding of the related data sources and relevant information is required for data scientists. Data formats can be categorized into (a)structured and (b)unstructured. Structured data can be easily stored, labeled, and represented in the tabular form. The tools used for structured data mostly include relational databases. Unstructured data include text, video, audio, and images which need higher structural organization for storing.

V. Recipients of Groundwater data

The groundwater data recipients include tthe following [13].

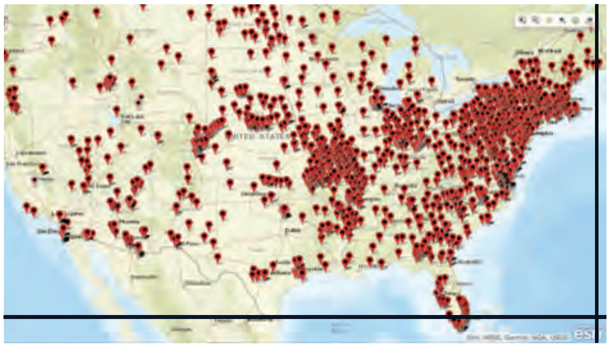


Fig. 2: U.S. Groundwater Information System.

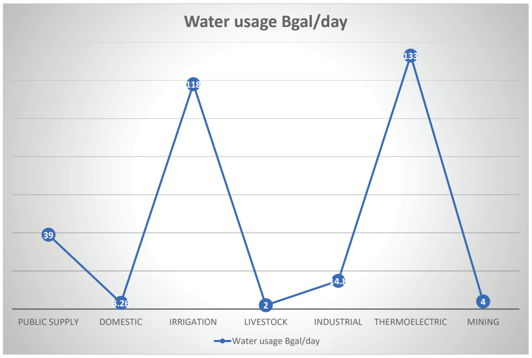


Fig. 3: Groundwater and Water Quality Data Users.

- Public Supply: They include water withdrawals by private and public distributors. Public suppliers provide water supply to a minimum of 15 connections and deliver to sectors ranging from domestic to industrial needs. A portion of the public water suply are used for parks, pools, wastewater treatment, and firefighting.
- Domestic: Domestic supplies are derived from individual wells at households. They are used for drinking, washing, watering gardens, etc.
- Irrigation: Agriculture is an essential means for food production and one of the major users of groundwater for irrigation in many parts of the world.
- Livestock: Water use for livestock in farms includes drinking water for livestock, dairy sanitation, and washing of the facilities for the animals and products.
- Thermoelectric: Turbines use water for thermoelectric power generation. Here, the water (from surface or underground sources) is circulated across heat exchangers back to the source. Electricity generation turbines use water for cooling at power plants. Most of the time it is a closed loop system.Through this recirculation, water losses are minimized.
- Industrial: Products manufactured for daily use of humans invlove water for their processing at the manufacturing sites, which contain metal, paper, oil, etc. are using groundwater.
- Mining: Water use in mining is primarily for dust control and other mining-related activities such as extracting different minerals from the ground and keeping the mine dry.

TABLE I: Data management Domains and Storage approaches in Smart Agriculture.

Application	Domain	Data storage	Security level	Cost	Computation
Nguyen et al. [4]	Supply-chain	Decentralized-OnChain storage	High-SingleHashing	High	High
Umamaheshwari et al. [5]	Crop Farming	Decentralized-OnChain storage	High-SingleHashing	High	High
Pincheira et al. [6]	Water usage	Decentralized-OnChain storage	High-SingleHashing	High	High
Turganbaev et al. [7]	Groundwater data	Centralized	Low-High Risks on Data	High	High
Yi et al. [8]	Groundwater data	Centralized	Low-High Risks on Data	High	High
Zhu et al. [9]	Groundwater data	Centralized	Low-High Risks on Data	High	High
Iwanaga et al. [10]	Groundwater data	Centralized	Low-High Risks on Data	High	High
G-DaM [Current-Paper]	**Groundwater data**	**Decentralized-OffChain storage**	**High-DoubleHashing**	**Low**	**Low**

VI. The Proposed Blockchain based Water-Quality Data Management System - A Broad Overview

With increasing groundwater contamination through petroleum storage tanks, septic systems, uncontrolled hazardous waste, nitrate pollution from overuse of fertilizers, hydraulic fracturing for oil extraction, and pharmaceutical products from waste water, measuring water quality is necessary. All the sources discussed in Section IV gather and store this information through their end systems or end stations. These end systems act as nodes and need to provide integrity at the storage level and security during transmissions. Each node combines storage and data allocating functionalities in this unified and synchronized mechanism for different files from various groundwater sources.

A. DDS-The Interplanetory File System (IPFS)

As discussed in the introduction about the limitations of blockchain holding large amounts of data, it is crucial to choose what information remains on-chain and what data should be off-chain. Some of the examples of off-chain storage include Storj1, FileCoin2, Sia3, and IPFS. All these use the same theory of distributing the files to different nodes by shredding and encrypting to provide security.

B. BC-Ethereum Smart Contract

One of the popular application tools for blockchain is Ethereum [14]. The transactions are through its cryptocurrency ether, and smart contracts are written for executing the logic. The programming language used to write the contract is solidity, compiled to generate bytecode that the Ethereum virtual machine (EVM) understands. The smart contracts are utilized for different purposes as they are Turing complete. The decentralization in Ethereum ensures that no nodes have execution control and establish trust based on the consensus mechanism. The data inside the transactions cannot be modified or altered with this mechanism. The programming language solidity comes with different access control functions, mappings, variables, and structures. The user can call these functions with the help of conditional statements, and if valid, the state is changed; if not, the state is reverted to its previous value.

C. Architecture

The IPFS distributed data storage stands between the groundwater source and BC to converse with the smart contract residing in the blockchain node. It mediates between traversing the transactions towards the Smart Contract's methods, administering its storage area, and interacting with the network and DHT. Fig. 4 demonstrates a comprehensive view of how data flow is taking place in our proposed system. Here the transactions are the data moving from IPFS to BC.

VII. The Proposed Algorithms for Blockchain based Framework

Algorithm 1 shows a detailed flow of the data from the end systems to IPFS and towards BC. The hashes are calculated for the ES files using public-key cryptography, SHA-256 inside DDS. The private key is picked up randomly to generate a corresponding public key and controls the access, which is a unique piece of information used to create the digital signatures and sign the groundwater data file. The FL denotes file from Endsytem (ES), FL (Buffer) is the file moved to buffer, and $FL_{265\ KB}$ mean file into 256 KB segments. The data file is combined with the private key to generate a code with the help of a private key. With the signatures, a secure hash message/hash string ”H (FL)” is digitally signed through IPFS, where H denotes a cryptographically secure Hash function. Once signed, the data gets called by the smart contract (SC) through the programming code function, set(). The contract implements an elliptic curve digital signature algorithm (ECDSA) signature on the ”H (FL)” to get the Signature output. Here, k represents the signing private-key of BC, m is the RLP encoded data. RLP-Recursive Length Prefix is an encoding technique used to serialize the ethereum objects. $F_{keccak256}$ is the keccak-256 hash function, $F_{signature}$ is the signing algorithm, and signature is the resultant signature in the given equation. After the data have successfully been signed twice, the smart contracts read and write to the BC using access policies.

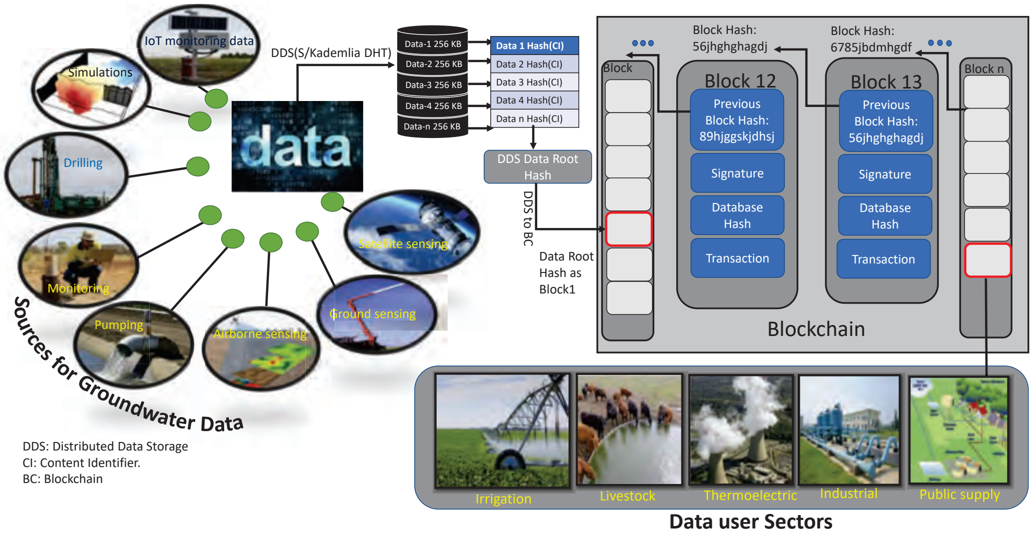


Fig. 4: Proposed Blockchain Architecture for Groundwater Data Management with DDS.

Algorithm 2 shows steps to the point for recovering data. The groundwater sector nodes must have the signature (r and s) values and serialized transactions to request the files. A signed message also contains the public key that belongs to the private key used to create the signature. The sector nodes get a signed data to perform signature authorization and check whether hash functions have been resolved. Only branches having these values can retrieve the file. Validating and solving the cryptographic puzzles have $O(1)$ complexity [15].

VIII. Implementation of Groundwater Data management System

Few dependencies are crucial to designing the current distributed storage application. A personal blockchain called ganache allows us to deploy smart contracts, develop the applications and run different tests. Ganache mirrors the actual public blockchain and allows local development. Once the ganache is running, an environment is set up to write smart contracts through NPM: node package manager and truffle framework (TF). NPM establishes local nodes, and the truffle suite provides tools to develop the current application. Some of the instruments in TF include smart contract management, automated testing, migrations and deployments of the contracts to the BC, managing network, console for development, script runner for Javascript client code, and client-side development [16]. The react-javascript (reactJS) framework is used for the front-end.

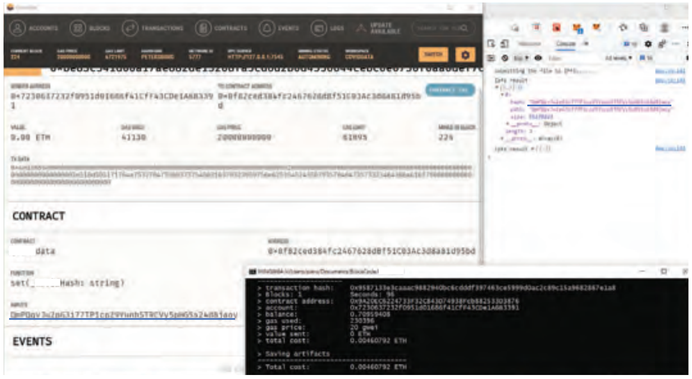
We tested the groundwater data management application through truffle on the local ganache blockchain to confirm its functionality. Once the contract code executes on actual BC, we cannot revert the code, hence the testing requirement. The current project is deployed onto the Ropsten test network to test in a live setting without real ether and mainnet tokens.

IX. Experimental Results

The connection between the user interface and the local ganache is established using metamask ethereum wallet. After the hash string is received, the application requires to confirm the transaction, and following approval, it gets stored on the BC, resulting in a new cryptographic hash through BC named Tx hash. The IPFS outcome and the ganache input are double-checked for valid hash strings sent to BC because of their irreversible nature. In Fig. 5, the output from IPFS and input for the ganache is indicated with an underline and is observed that the sender address in ganache is the same account address in deployed ropsten.

Once data is verified through ganache, we deploy DDS hash on to real scenario blockchain explorer ropsten testnet, which mirrors the actual mainnet. Adopting to explorer delivers the details of the authentic BC transaction hash with Tx fees and time. On the testnet, specifics such as Tx hash, status, block number, Timestamp, addresses, ether, Tx fee, price of the ether, gas limit, and nonce, along with input data are provided. The ropsten testnet is presented in Fig. 6. Note that the beginning source address is similar everywhere in our transaction. Once the block gets created, we check its height, mining time, and rewards for the block illustrated in Fig. 7.

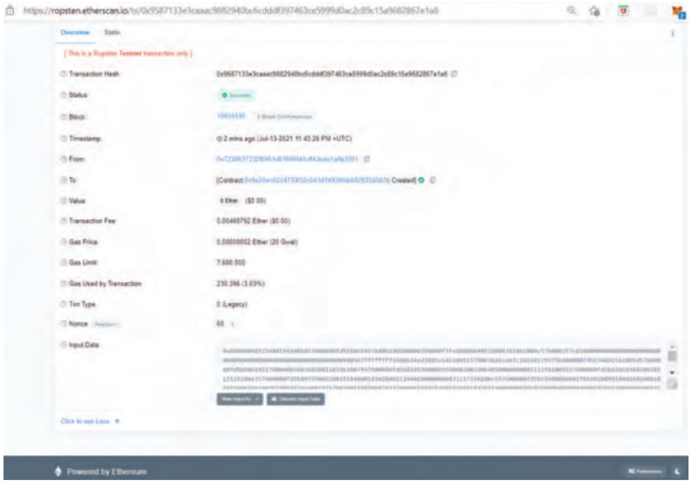
We used datasets from two sources, one from Kaggle [17], and another from Nebraska groundwater quality clearinghouse data [18]. The end system has groundwater data from multiple sources. The information sets we tested include different file types of .csv, .zip, .txt, .gis, etc., of various sizes. From the daily Ycharts tool of Ethereum, we monitored the price of one ether to be $1811.41 and mining time as 13 sec for 1MB data [19]. It takes 0.032 ether fees for 1 KB of information to get uploaded onto BC [19]. With all these statistics gathered, we calculated the values for data directly to BC and compared



```
SENDER ADDRESS                                TO CONTRACT ADDRESS
0x7230637232f0951d01686f41CfF43CDe1A683391    0x8f82ced384fc2467628d8f51C03Ac3d8a81d95bd
VALUE      GAS USED   GAS PRICE      GAS LIMIT   MINED IN BLOCK
0.00 ETH   41130      20000000000    61895       224

CONTRACT
CONTRACT                ADDRESS
data                    0x8f82ced384fc2467628d8f51C03Ac3d8a81d95bd
FUNCTION
set(_____Hash: string)
INPUTS
QmPQqv3u2p63177TP1cp29YunhSTRCVy5pHG5s24dBjaoy

EVENTS
```

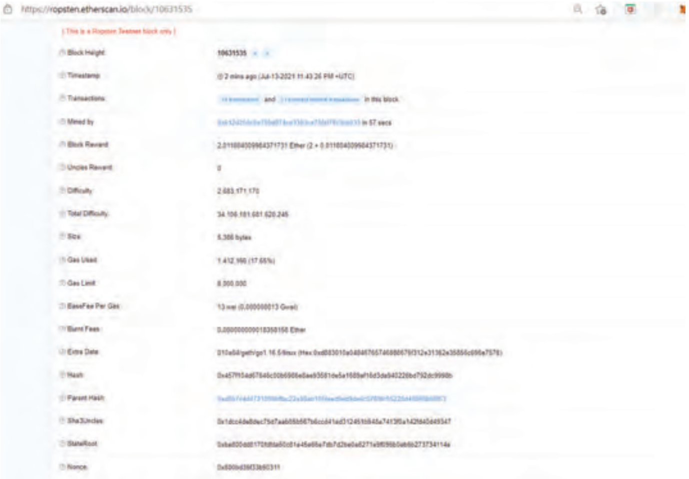
```
> transaction hash:    0x9587133e3caaac9882940bc6cdddf397463ce5999d0ac2c89c15a9682867e1a8
> Blocks: 1            Seconds: 96
> contract address:    0x9A20EC6224733F32C843D74938Fcb88253303876
> account:             0x7230637232f0951d01686f41CfF43CDe1A683391
> balance:             0.70959408
> gas used:            230396
> gas price:           20 gwei
> value sent:          0 ETH
> total cost:          0.00460792 ETH

> Saving artifacts
-------------------------------------
> Total cost:          0.00460792 ETH
```

Fig. 5: Hash String verification in local ganache and deployment to Rospsten Testnet.



```
https://ropsten.etherscan.io/tx/0x9587133e3caaac9882940bc6cdddf397463ce5999d0ac2c89c15a9682867e1a8
Overview   State
[ This is a Ropsten Testnet transaction only ]
Transaction Hash:           0x9587133e3caaac9882940bc6cdddf397463ce5999d0ac2c89c15a9682867e1a8
Status:                     Success
Block:                      10631535   1 Block Confirmations
Timestamp:                  2 mins ago (Jul-13-2021 11:43:26 PM +UTC)
From:                       [illegible]
To:                         [Contract [illegible] Created]
Value:                      0 Ether ($0.00)
Transaction Fee:            0.00460792 Ether ($0.00)
Gas Price:                  0.00000002 Ether (20 Gwei)
Gas Limit:                  7,600,000
Gas Used by Transaction:    230,396 (3.03%)
Txn Type:                   0 (Legacy)
Nonce:                      60
Input Data:                 [illegible]
Click to see Less
Powered by Ethereum
```

Fig. 6: Transaction Hash.



```
https://ropsten.etherscan.io/block/10631535
[ This is a Ropsten Testnet block only ]
Block Height:        10631535
Timestamp:           2 mins ago (Jul-13-2021 11:43:26 PM +UTC)
Transactions:        [illegible] and [illegible] in this block
Mined by:            [illegible] in 57 secs
Block Reward:        2.011604099984371731 Ether (2 + 0.011604099984371731)
Uncles Reward:       0
Difficulty:          2,683,171,170
Total Difficulty:    34,106,181,681,620,245
Size:                5,388 bytes
Gas Used:            1,412,996 (17.66%)
Gas Limit:           8,000,000
BaseFee Per Gas:     13 wei (0.000000013 Gwei)
Burnt Fees:          0.000000000018358158 Ether
Extra Data:          [illegible]
Hash:                [illegible]
Parent Hash:         [illegible]
Sha3Uncles:          [illegible]
StateRoot:           [illegible]
Nonce:               [illegible]
```

Fig. 7: Mining Time.

Algorithm 1 Data from Groundwater endsystems to IPFS and BC.

```
1: ES, BC have their own Public and Private Keys (PuES, PrES) and (PuBC, PrBC) respectively
2: ES(FL)⟶FL(Buffer)⟶FL $_{265\ KB}$.
3: SC[set()] ⟶FL $_{265\ KB}$⟶DDS.
4: File cryptographically hahed with SHA 256 to give unique fingerprint CI(Content Identifiers).
5: PuES= H(PrES * G), where G is the constant ,* is the mathematical operation performed in one direction and H is the secure hash function.
6: if FL==H(PrES * G)==H (FL) then
7:     Publish H ( FL) ⟶DDS, using IPFS client.
8:     SC[get()] and SC[Publish()] functions to publish " H (FL) " from DDS.
9:     Sign "H (FL) " with ECDSA, Signature = F $_{signature}$ (F $_{keccaK256}$(m),k).
10:    Append ECDSA signature to the transaction.
11:    if "H(FL)" signed with ECDSA then
12:        Smart contract hash maps are used to access IPFS hashed string to ethereum accounts .
13:        Device provides Owners, Address and Device ID as key in Hash map along with encrypted Hashed string to be written on Blockchain.
14:        The Hash of the Encrypted Data is published on to Smart contract function and checks validity of the data with write access policy.
15:        if Device owner, Address corresponds to device ID. then
16:            Execute the Write operation.
17:        else
18:            Discard Write operation.
19:        else
20:            End the Process.
21:        else
22:            End the Process
23:        end if
24:    end if
25: end if
26: Repeat the steps from 1 through 25 every time end system collects groundwater data.
```

them with data moving to BC through DDS. Table II illustrates the reduced cost, fees, and time intervals for the same datasets.

X. Conclusion and Future Direction

In this work, we have addressed a novel DDS and BC approach for groundwater data management to solve the problems related to data integrity, privacy, data quality, and latency. Moreover, it displays the power of DDS and BC to deliver solutions at the architecture level for the discussed issues. The current design makes use of public blockchain with smart contracts having control over access and data. In future work, the stakeholders and groundwater sectors can be made

TABLE II: Comparing Mining-Time, Gas-Fee and Tx-Cost for Groundwater dataflow between only BC and BC with DDS.

File	File-Size	Deploy-Time (Sec)	Mining-Time (Sec)[BC]	Mining-Time (Sec) [BC+DDS]	Gas-Fee[BC]	Gas-Fee [BC+DDS]	Tx-Cost [BC]	Tx-Cost [BC+DDS]
.txt-chemicaluseinagriculture	97 KB	32	13	39	3.104eth	0.00460792eth	$5,622	$8.34
.csv-wateruseinagriculture	4.41 MB	24	57	77	141.12eth	0.00489103eth	$255,626	$8.85
.csv-affectedwaterbodies	4.97 MB	4	64	7	159.04eth	0.00491564eth	$288,086	$ 8.9
.zip-Nebraskagroundwaterdata	11.6 MB	72	150	46	371.2eth	0.00367895eth	$672,395	$6.66
.gis-Waterdataset	52.7 MB	96	685	57	1686.4eth	0.00543623eth	$3054,761	$ 9.8

1 Eth=1811.41 Dollars, 1KB=0.032 Eth, 1MB=32.768 Eth

Algorithm 2 BC to Groundwater data User sectors.

```
1:  BC and US have their Public and Private Keys (PuBC, PrBC) and (PuUS, PrUS) respectively.
2:  The data access request is sent by Requester.
3:  Data access request is signed by Requester's private key (PrR) and signature is appended along with request data.
4:  Data access request along with the signature is encrypted by public key of End system (PuES) and published through Smart contract Client Program.
5:  The Request is decrypted by ES and verifies the message integrity using the signature.
6:  if Match in signature then
7:      The Requester will request for permission to read the data.
8:      The Requester will provide Device Owner, Device Address and Device ID.
9:      Smart contract maintains a Hash map that contains Device owner, Device address, Device ID as key along with User sectors registered.
10:     if Requestor's Device Owner, Device Address and Device ID matches Smart Contract Hash map then
11:         Requester can access the data to read.
12:     else
13:         Data Access Denied.
14:     else
15:         End the process.
16:     end if
17: end if
18: Repeat the steps from 2 through 17 every time there is a new user sector access request.
```

confidential through private BC for more extensive control of the groundwater data flow.

References

[1] "Groundwater," 2021, last Accessed on 14 July 2021. [Online]. Available: http://www.waterencyclopedia.com/Ge-Hy/Groundwater.html

[2] P. Fitch, B. Brodaric, M. Stenson, and N. Booth, *Integrated Groundwater Data Management*. Springer International Publishing, 2016, pp. 667–692.

[3] D. Puthal, M. S. Obaidat, P. Nanda, M. Prasad, S. P. Mohanty, and A. Y. Zomaya, "Secure and sustainable load balancing of edge data centers in fog computing," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 60–65, 2018.

[4] D.-H. Nguyen, N. H. Tuong, and H.-A. Pham, "Blockchain-based farming activities tracker for enhancing trust in the community supported agriculture model," *IEEE Information and Communication Technology Convergence*, October 2020.

[5] S. Umamaheswari, S. Sreeram, N. Kritika, and D. R. Jyothi Prasanth, "BIoT: Blockchain based iot for agriculture," *IEEE Advanced Computing*, May 2020.

[6] M. Pincheira, M. Vecchio, R. Giaffreda, and S. S. Kanhere, "Exploiting constrained iot devices in a trustless blockchain-based water management system," *IEEE Blockchain and Cryptocurrency*, August 2020.

[7] E. Turganbaev, S. Rakhmetullina, Z. Beldeubayeva, and V. Krivykh, "Information system of efficient data management of groundwater monitoring the republic of kazakhstan," *IEEE Application of Information and Communication Technologies*, October 2015.

[8] Z. Yi, L. Xiaodong, L. Jiping, and Z. Yu, "The design and applications of services platform system for water data basing on webgis," *IEEE Information Management and Engineering*, April 2010.

[9] Z. Yunqiang, Z. Shaoxia, and Y. Mengliang, "Study on groundwater data sharing based on metadata," *IEEE International Geoscience and Remote Sensing Symposium*, vol. 2, July 2005.

[10] I. Takuya, E. S. Sondoss, and J. Anthony, "Design and implementation of a web-based groundwater data management system," *Elsevier*, vol. 93, July 2013.

[11] X. Yushu, K. Hoyoung, and W. Michelle, "Developing county level data of nitrogen fertilizer and manure inputs for corn production in the united states," *Elsevier*, vol. 309, August 2021.

[12] "Grounwater data," 2021, last Accessed on 11 July 2021. [Online]. Available: https://maps.waterdata.usgs.gov/mapper/nwisquery.html

[13] "Who uses groundwater?" 2015, last Accessed on 14 July 2021. [Online]. Available: http://gwhub.srw.com.au/who-uses-groundwater

[14] S. Vangipuram, S. Mohanty, and E. Kougianos, "CoviChain: A blockchain based framework for nonrepudiable contact tracing in healthcare cyber-physical systems during pandemic outbreaks," *SN Computer Science*, vol. 2, September 2021.

[15] I. Baumgart and S. Mies, "S/Kademlia: A practicable approach towards secure key-based routing," *IEEE Parallel and Distributed Systems*, December 2007.

[16] M. A. Andreas and W. Dr.Gavin, *Mastering Ethereum*. O'Reilly, December 2018.

[17] kaggle, "Datasets," last Accessed on 14 July 2021. [Online]. Available: https://www.kaggle.com/datasets

[18] "Nebraska groundwater quality clearinghouse," last Accessed on 14 July 2021. [Online]. Available: https://clearinghouse.nebraska.gov

[19] "Ycharts," 2009, last Accessed on 14 July 2021. [Online]. Available: https://ycharts.com/indicators/ethereum_price