Automating GDPR Compliance using Policy Integrated Blockchain

Abhishek Mahindrakar and Karuna Pande Joshi
Information Systems Department
University of Maryland Baltimore County
Baltimore, MD, USA, 21250
Email: { amahindrakar, karuna.joshi}@umbc.edu

***Abstract*—Data Protection regulations, like GDPR, mandate security controls to secure Personal Identifiable Information (PII) of the users which they share with service providers. With the volume of shared data reaching exascale proportions, it is challenging to ensure GDPR compliance in real time. We propose a novel approach that integrates GDPR Ontology with Blockchain to facilitate real time automated data compliance. Our framework ensures data operation is allowed only when validated by data privacy policies in compliance with privacy rules in GDPR. When a valid transaction takes place the PII data is automatically stored off-chain in a database. Our system, built using Semantic Web and Ethereum Blockchain, includes an access-control system that enforces data privacy policy when data is shared with third parties.**

Keywords: Ontology; General Data Protection Regulation; Big Data; Privacy Policy; Blockchain; Smart Contract; Ethereum.

I. INTRODUCTION

Service Providers are increasingly collecting personal details of their consumers to determine user-behavior patterns associated with market trends, fraud detection or forecasting customer loyalty. Most of these datasets include Personally Identifiable Information (PII) data [34] which are used for big data analytics. This large volume of PII data is often managed and shared using Cloud-based services. Secure usage of this data is critical to prevent potential for inappropriate dissemination of an individual's private information. Many regulatory and standard bodies have released data protection regulations like the European Union's General Data Protection Regulation (GDPR), US Health Insurance Portability and Accountability Act (HIPAA), Payment Card Industry Data Security Standard (PCI-DSS), etc. to ensure correct private data usage[32][36][37]. It is the responsibility of the providers to adhere to these data protection regulations and ensure the security and privacy of the consumer data they collect.

Providers acquire the consent of their consumers to use and share their PII data via legal contracts like terms of services and privacy policy documents. Currently, these privacy policies are text-based documents containing legal jargon that requires significant manual effort to parse. The data protection regulations are also text-based documents which often cross-reference different sections in the same document or other laws. Hence, it is difficult to determine in real-time if an operation on the data will result in privacy violation. To automate the process of validating the privacy policies they should be made machine-readable.

We have developed a novel approach to capture the knowledge embedded in the policy and regulatory documents in the form of semantically rich knowledge graphs that is machine-processable and can be reasoned over to automatically validate every data usage operation. This was accomplished using Semantic Web technologies, Natural Language Processing (NLP) and Text Mining. While we are creating knowledge representations of privacy policies and regulations, for this paper we have considered only the GDPR regulation. We have integrated our GDPR knowledge graph [31][32] with Blockchain technologies to have an audit log of every operation and the corresponding GDPR policy that permits the operation. Every data operation is validated against the rules defined and depending on the validation of the transaction, the blocks are created to store transaction data. This system is private and permissioned as it includes Semantic Web, Ethereum and encryption techniques. Encrypted datasets are distributed over the network, this dataset does not reveal the personal details of the consumer and hence helps in avoiding the breach of data.

Contribution : Extending our previous work [33], we parse knowledge graph for transaction authentication and create blocks over blockchain network. Before a transaction is invoked, a SPARQL query is executed which checks with the privacy ontology for permissions. The participants are initially registered on the network using ganache-cli. Ganache-cli is a private blockchain network for Ethereum development. Once the consumers and providers are registered on the network, transfer of data is permitted. For every successful transaction, a block of data is created with Transaction Hash, Block Number, Gas usage , Block Time and Result. As blockchain is an emerging technology and the process of storing and retrieval of data is much different than the mature enterprise databases, complying with all regulations is difficult to achieve. Hence, we plan on extracting and storing block data in an external encrypted file.

For future references, if the user desires to know the details of the transactions, the user needs to have the public key to decrypt the file containing the Transaction Hash. Once the Transaction Hash is obtained the user can query the blockchain and retrieve results.

In this paper we describe our approach in detail and include the results of our initial implementation. Section II describes the background and related work. In Section III, we describe the methodology and architecture of the system. In Section IV we describe the experimental results. Section V has conclusion and Section VI covers our ongoing work.

II. RELATED WORK

In a service environment, consumers and providers need to be able to exchange information, queries and requests with some assurance that they share a common meaning. This is critical not only for the data but also for the data protection policies followed by service consumers or providers[29]. Nasr Al-Zaben et al.[11] proposed a data tracking system where the user should have the knowledge of the location of data consisting of PII which can be used to easily track the user information. They also defined Potential Personally Identifiable Information where the information may be used to identify an individual. The paper also proposed an architecture where the user shares hash values of PII, NPII (Non-Personally Identifiable Information) and PPII (Potential Personally Identifiable Information) to the controller and similarly the controller shares it to the processor. A block is created on a successful completion of transaction [11]. A list of PII are mentioned below in Table 1 which were illustrated by Joshi et al. [15]. Zyskind et al. [16] described a decentralized personal data management system where users can track and control their data. Their system can be used to perform instructions like storing, querying and sharing of data. The type of transactions which are used by their system is Taccess for access management and Tdata for storage and retrieval of data.

Table 1: Personally Identifiable Information

Personally Identifiable Information (PII)
Personal Details, Address, Phone, Personal Identification Number, Personal Characteristics, Property Details, Computer Assets, Geographical Indicators, Employment Information, Medical Information, Education Information, Financial Information

A. Ethereum Blockchain

Blockchain is a peer to peer distributed ledger technology which is being used in many scenarios which relates with trustable, confidential, secure and auditable platform. We have used Ethereum, which is an open source software used to build our architecture. The Ethereum framework is used as a protocol for business to business or business to customer transactions[25]. A useful feature of blockchain over other technologies for storing transactions and its data is non-repudiation [23].This feature can be used to verify the transactions using digital signatures and public key infrastructures [14]. Ethereum is a decentralized blockchain platform which runs smart contracts. Decentralized applications are executed on this platform. The two types of accounts on Ethereum network are Externally Owned Accounts which are controlled by private keys and Contract Accounts controlled by the code in the contract.[19] The Smart Contract refers to the code in the contract. The code in the contract is executed only when a transaction is invoked. The process of mining includes adding a block to the network. This happens after every successful transaction. Only members which are registered on the network are allowed to initiate mining. The block will be discarded when the authentication process fails. Ganache-cli is an Ethereum client that connects with local decentralized applications. To write transactions over blockchain network we have used Truffle framework. Truffle is the most popular Ethereum framework to create and compile solidity contracts. Truffle uses solidity compiler which can be used to compile and deploy smart contracts [34]. Everything we write to a blockchain network is permanent it can't be removed and hence we have used ganache-cli which is very useful for testing and development. Ganache-cli is an in-memory based virtual blockchain. Once the ganache is closed all the accounts will be destroyed. A local ganache network creates 10 accounts by default and their private keys, it creates mnemonic for that particular network. The network can be initialized on any listening port.

When it comes to security it is comprised of three components: Confidentiality , Integrity and Availability. Blockchain satisfies all the three components. 1. Confidentiality means the state of keeping an entity secret; all the blocks in the blockchain are encrypted and stored in hash format; it satisfies confidentiality. 2. Integrity implies the accuracy, completeness and consistency of data throughout its lifecycle; blockchain is immutable, once data is stored on the chain it cannot be changed and it stays throughout the chain's lifetime [14]. 3. As blockchain is decentralized, the storing of data is not dependent on any single block or peer. This way blockchain is always available and the system does not need to rely on any third-party service.

B. Smart Contract

Smart contract is created to interact with blockchain. The smart contract cannot be changed or manipulated once deployed. When the smart contract is deployed three interfaces are loaded - init, invoke and query. Init is used to initialize the contract. Query is initiated when the end users query the blockchain. Invoke is called when the user wants to input data on blockchain. When the smart contract is compiled two variables are created – ABI and Bytecode [24]. In smart contract, a get represents a query to retrieve information about the current state of a business object, a put

is used to create a new business object or to modify an existing object in the ledger world state and a delete represents the removal of a business object from the current state of the ledger, but not its history[27]. Critically, in all cases, whether transactions create, read, update or delete business objects in the world state, the blockchain contains an immutable record of these changes.

Smart contracts are referred to as chain code, this chain code helps you to read and update data on the blockchain ledger. Smart contracts are invoked from peers on the ledger and then instantiated on channels[22]. All the members of the blockchain who want to submit transactions or read data by using smart contracts need to install the contract on the peer. The smart contract is defined by name and version of the contract. Only one network member needs to instantiate the smart contract. If a peer with a smart contract installed joins the channel where the same contract is already instantiated, the smart contract container starts automatically [29]. A smart contract defines the rules between different organizations in executable code. That's because a smart contract can implement the governance rules for any type of business object, so that they can be automatically enforced when the contract is executed. For example, a smart contract can ensure that a car is delivered in a specific time frame or a set of transactions are executed on prearranged terms and conditions. Most importantly, however, the execution of a smart contract is much more efficient than a manual human business process[30]. Emanuel et al. [29] proposed a smart contract language that is human readable and also binds all the legal conditions. They coined the domain-specific language as SmaCoNat with illustrated examples to show that language enhances readability and safety without violating any natural language specifics. Chun-Feng Liao et al. [29] presents a service platform that helps in the development, testing, and deployment of smart contracts.

Decentralized App: Apps which use smart contracts are Decentralized Applications (Dapp). The goal of Dapp is to have a user interface to your smart contract. These apps can be run on central servers or can also be executed on top of a peer node. Only those accounts that are registered on the network have access to the smart contract and the data transacted, preserving the privacy and confidentiality of both[30].

Reasons to choose Ethereum over other Blockchains:

a. Ethereum is a public blockchain network and focuses on running programming code on decentralized application and smart contracts through its virtual machine (EVM).
b. By querying to Ethereum, we can retrieve results of a block using the block hash or transaction hash.
c. The communication between consumer and provider is efficient and it takes seconds to create a block.

C. General Data Protection Regulation (GDPR)

The GDPR is recently proposed by European Union as a privacy regulation. Whenever any personal data is shared between two parties the transaction should comply with the rules and regulations of GDPR. The law defines that personal data processing is prohibited unless it is allowed by law or has consent of the owner [10]. The data owner also has the right to invoke the consent any time after the data is shared. The withdrawal of permission of personal data must be easy and the data should be erased from all the locations. According to GDPR, all the data which is processed in our system will be encrypted and stored on the database. This minimizes the risk of cyber-attacks and data leakage. The personal data can be identified as the data which directly identifies a person or an entity. The personal data can be either subjective like name, address, contact number or objective like opinions, suggestions and expressions using which a person can be corelated. The processing of personal data should also be taken place in compliance of the rules and regulations of the contract.

All the details related to the processor, controller and categories should be defined. The right to be forgotten regulates the erasure obligation. The controller and the processor are obliged to delete the personal data when processing is complete, and when the data is no longer required. Lavanya et al. developed a data compliance ontology which integrates the knowledge representation of GDPR[31][32]. Semantic Web Ontology Language is used by the author to create a knowledge graph that will be consumed by our proposed system for implementation. The GDPR rolled out regulations related to sharing and storing of personal data with respect to the consumers and providers. The GDPR does not care about where the data is stored but the data providers and consumers must follow regulations.

The GDPR rules which are considered for consumer and provider for our system [35]:

1. **Processing of personal data**: The personal data collected should be stored and processed in a transparent manner. The collected data should not be used for purposes not defined in the policy document. Under the Rights of the EU Data Subjects , Know Your Customer(KYC) is defined wherein personal data is consumed for processing purposes.
2. **Responsibility of Consumer**: The consumer should be responsible enough to implement methods which are in accordance with the regulation.
3. **Data Protection and by Default**: The organizations should collect and store data only for purposes mentioned in the policy document and every transaction should be recorded. The data used in the application should be pseudonymized, anonymized or encrypted.
4. **Responsibility of Provider**: The provider should hold onto confidentiality obligations and also follow the rules while appointing sub-providers. The lawful consent of the provider and sub-provider should be followed to track and manage data between data subjects, processors and controllers.
5. **Processing under consumer authority**: The consumer and provider should be aware of the data provenance.

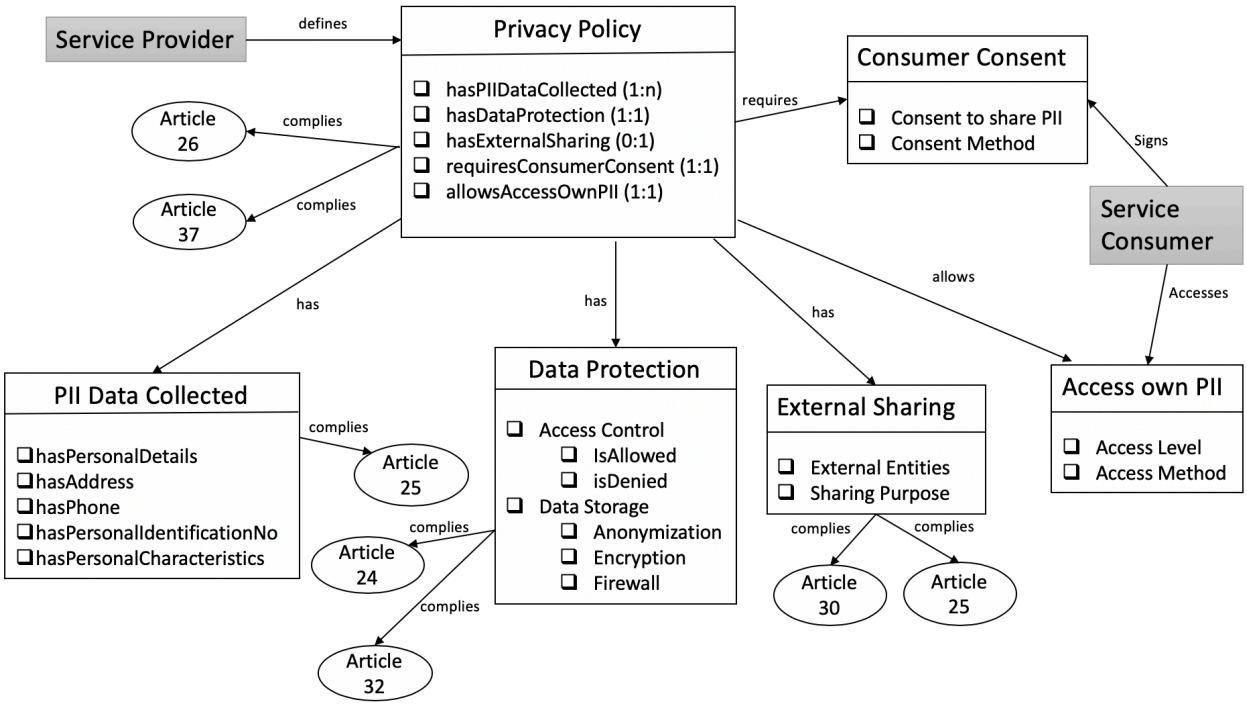

Figure 1: Snapshot of the Privacy Policy Knowledge Graph for our System

III. METHODOLOGY

Privacy Policy Ontology

Joshi et al. [15] proposed classes of an ontology which can be used as components in the system. The author recognized that there should be at least one instance of consumer and provider consent to access the PII data. Figure 1 describes the ontology classes consumed by our system.

The classes are described below:

Data Protection: The class defines the access control and data storage policies. The rules defined in the access control class are isAllowed or isDenied; this class is based on the policy definition of the privacy policy and sets the attribute. Date Storage class defines the method in which the data is stored which can be either anonymized data, encrypted data. The Firewall in the data protection class can also define which all IP's are allowed to access the data.

PII Data Collected: The attributes which cover the personal data include name, address, telephone number, etc. Many other details are included in PII data as it might reveal the user information like the insurance details, financial and education details.

External Sharing: This class defines the policies defined for sharing the data externally and the purpose related to it.

Consumer Consent: The privacy document should explicitly mention the storing and maintaining of the PII data acquired from the consumers. The permission to share this data should also be approved by the consumer before the provider shares it.

Access own PII: Consumers should be able to access their PII data. Joshi et al. [15] also covered various standards and guidelines for data privacy policy by multiple organizations like the National Institute of Standards and Technology(NIST), European Union Data Protection Standards, etc. The key points identified by the author are 1. Explanation of how businesses collected and used personal information, 2.Age restriction of collecting data, 3. Make new customers aware of the privacy policy and data control, 4. Purpose of the data collection, 5. How is the data which is collected secure, 6. How users can access their collected data.

In this paper, we have used Semantic Web, Natural Language Processing techniques, Ethereum and Advanced Encryption Standard (AES) to automate the process of data transfer between either consumer to provider or provider to provider in compliance with GDPR. Our application retrieves privacy policy and GDPR ontology from the cloud storage. As we are aware that blockchain technology is not compliant with GDPR we are using blockchain; smart contracts to store, track and authorize transactions. GDPR Article 25 - Data protection by design and default involves encrypting data which is followed by our system. Article 24 - Responsibility of the Controller, it is to process data in accordance with the regulation, our system makes sure all the transactions are in compliant with the policy ontology. Article 30 - Records of Processing Activities, as blockchain is immutable, all the transactions (valid, invalid) are stored on the network. Article 37 – Designation of Data Protection Officer (DPO), the trusted controller in our system covers the responsibilities of a DPO[10]. We have identified a few steps in our process and detailed them below. Figure 2. shows the architecture of the system.

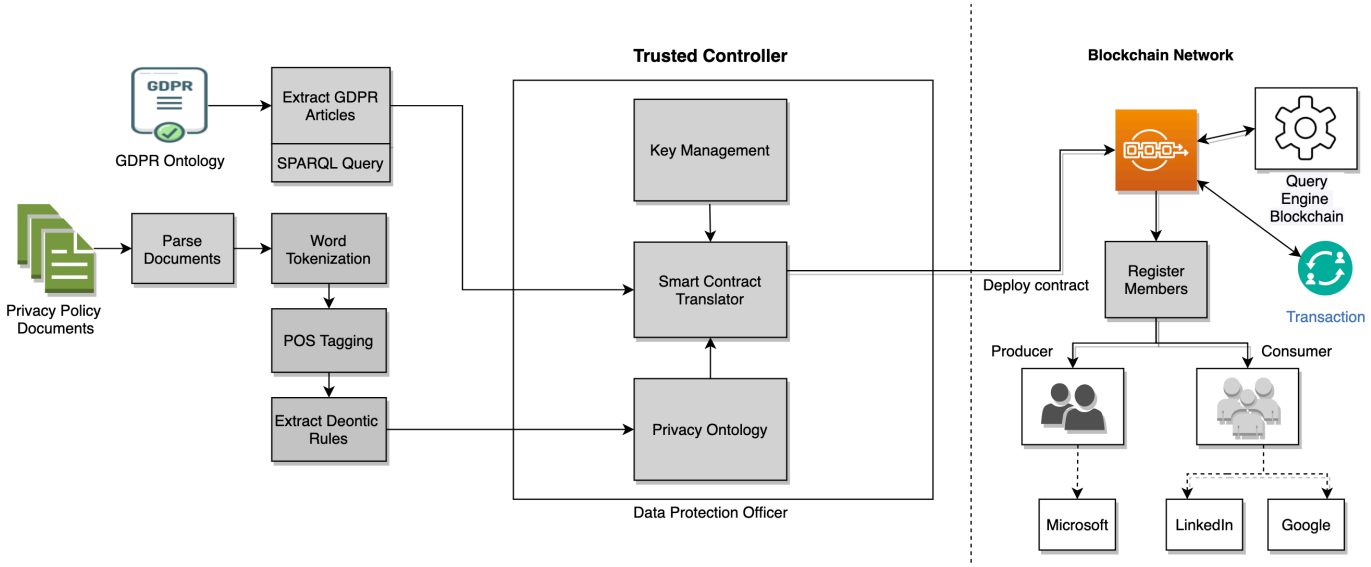

Figure 2: Architecture of the System

Key Modules of the System:

Parse Privacy Policy: The privacy policies are consumed from either online or on local repository. These privacy policies are in text format and they are converted into owl for classification[18].

Extract Key Terms and Rules: The ontology is loaded to perform classification and also to create new instances for further implementation. The classes in the ontology can be added or removed. A relation consists of tuples which are subject, property and object. The documents are tokenized, then they are tagged based on part of speech like noun, pronoun, adjective, verb, adverb, etc. The deontic words are extracted from every sentence to create a deontic rule which will help the system in defining if the sentence claims to be a permission, obligation, dispensation or prohibition with respect to the subject. Obligation can be defined based on the words – should, shall, must. Permission and Dispensation based on the words – can, may, could. Prohibition can be based on the words – can't, must not, may not. The rules are extraction for Permission and Prohibition. The following grammar rules are used for classification [35].

Permission:

< Noun/Pronoun > < deontic > < verb >

Prohibition:

< Noun/Pronoun > < deontic > < negation > < verb >

Extract GDPR Articles: A GDPR ontology knowledge graph is created and only articles related to consumer and provider are extracted using a SPARQL query. SPARQL is an RDF query language, it is a semantic query language used to extract details from an ontology. The query extracts the articles which are further used for the reasoning of the smart contract translator.

Trusted Controller :

Key Management: Public key infrastructure is used in blockchain technology which helps in authenticating the network and ensure integrity. We use Ganache, previously known as TestRPC, this sets up 10 default Ethereum accounts with their private keys and preloads them with 100 simulated Ether each.

Privacy Ontology: Once the extraction of key terms and rules is completed a privacy ontology is created which is referred by all the future transactions. Every transaction needs to be compliant with the privacy policy regulations and hence when any transaction is executed the initial step is to check if it has the permission. Once the permissions are checked, the transaction is successfully executed.

Smart Contract Translator: A smart contract is nothing but a computer protocol that has methods defined based on the access policies. The contract has methods that are used to execute an action on the blockchain. If the action is not defined in the contract, the transaction is failed. Every transaction the system calls a method on the smart contract which verifies if the action is permitted.

Register consumer/provider on blockchain: The stakeholders are registered on the blockchain network and only they can initiate a transaction. The registered members are assigned an account number and a private key. This account number is unique to the member and is used for all the transactions. There can be multiple users on the network as long as the unique account numbers and private keys are valid.

Write on Blockchain: Once a transaction is successful the result of the transaction is used as an input to create a block over the chain. The block consists of transaction details and a hash. All the transactions which are executed on the chain are immutable.

Hence nothing can be deleted or modified once it is written. The blocks are protected, the transaction receipt can only be revealed when the user is aware of the transaction hash or the block hash.

Extract Data off-chain: The PII data on the blockchain network is immutable. It cannot be updated or deleted. Hence, we extract the transaction data after every transaction and store it in an encrypted file. The file is further stored on a database. We propose storing data off-chain to accomplish the GDPR rule. The rule 'Right to be Forgotten' can be partially achieved if we delete the encrypted copy of the data or deny sharing of public key with sub-providers, the provider will not have the transaction details or the data of the consumer.

Encrypt Data and Store data: The text data which is extracted from the transaction is stored in an encrypted format. The encryption of file is executed with aes256 and then stored on the database.

Query Blockchain: The blockchain can be queried to view the results of transactions. The specific transactions can be retrieved using the transaction hash or block number. The query uses the transaction id or transaction hash to retrieve the transaction receipt. The queries use web3 libraries to retrieve details on blockchain. The examples of the query statements:

web3.eth.getBlock(blocknumber,console.log);
web3.eth.getTransactionFromBlock(Transaction Hash);

IV. EXPERIMENTAL RESULTS:

Experiments were performed on a single Linux server – RedHat Version - CentOS Linux release 7.6.1810 (Core), Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz. Components used are – Ganache CLI v6.9.0, Truffle v5.1.12, Ethereum web3, Solidity v0.5.16, Node v11.13.0, Web3.js v1.2.1. The technologies used are – python version 3.6.7, Nodejs version 6.16, Go version 1.12.1, Docker version 18.09, NPM version 6.13.7, OpenSSL 1.0.2k-fips.

Privacy policies are consumed in text format. The word tokenization, part of speech tagging and deontic rules extraction is established using NLTK python library. A privacy policy ontology is created which is used for verification of transactions on the blockchain network. Relations are added or removed depending on the requirements of the use case. The transactions executed on the network should be policy compliant. The PII data is encrypted using a private key, AES is used for encryption of data and the file is stored on the database. Ethereum blockchain network is then queried to retrieve results as per the user requirements. If the transaction fails to transfer data one of the reasons for failure can be the user is not registered on the network which implies user is not authorized for any transactions on the network. In our system, all the consumers and the providers have to be registered on the blockchain network for initiating transactions. As in blockchain the data is always updated as per the consensus algorithm and the data values are matched according to the ledger value, keeping the data off-chain helps in tracking the previous values.

All the transactions take place over blockchain network, the data will always be tracked and the end users will be aware of where the data is being shared and to how many providers it is being transferred.[17] With this setup of decentralized platform, transferring assets, registering users, controlling data related transactions in compliance with regulations is much simpler and easier.

To set up the environment we created a blockchain network using ganache-cli. Ganache-cli creates 10 accounts which were then assigned to our consumers and providers. Figure 4 shows the accounts created by ganache-cli on Ethereum network. All the members on the network need to be registered on the network using the account id and private key. When the network is up, the smart contract is deployed which is shown in Figure 4.The advantage of using a smart contract is, once the contract is deployed by a single instance on the network it is deployed on the ledger and can be accessed by all the members on the network, hence all members will have an updated version of the smart contract. The smart contract is immutable and hence it confirms security over the network[30]. The important parameters to be noted are the transaction hash, block number, and contract address.

Once the contract is deployed, we can test the contract methods using the test feature of truffle. Truffle is the Ethereum framework to create, compile and deploy solidity contracts. Truffle automation tests use the Mocha framework to test solidity contracts. Mocha is a feature-rich JavaScript test framework. Figure 5 shows the execution of the DataComplianceContract which is tested on our network once the solidity contract is deployed. The test execution shows us the time(in milliseconds) required to execute each method in the contract. All the contracts and their methods mentioned in the test cases will be executed with sample variables[26].

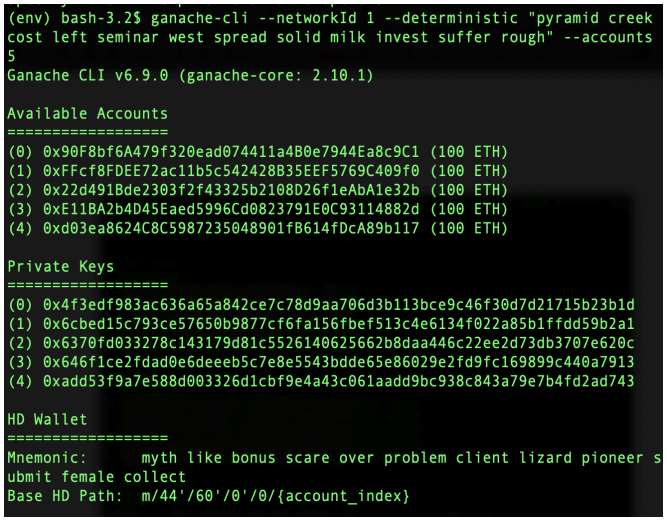

Figure 3: Accounts created by Ganache-cli

```
> transaction hash:    0xc8d7e1673ad64ce4554babf0ea43c2f372adbc531d3fa630a2f3c5fdbf4d1ff8
> Blocks: 0            Seconds: 0
> contract address:    0xCfEB869F69431e42cdB54A4F4f105C19C080A601
> block number:        3
> block timestamp:     1585064761
> account:             0x90F8bf6A479f320ead074411a4B0e7944Ea8c9C1
> balance:             99.93982408
> gas used:            2794411
> gas price:           20 gwei
> value sent:          0 ETH
> total cost:          0.05588822 ETH
```

Figure 4: Result of Smart Contract Deployment

```
(env) bash-3.2$ truffle test
Using network 'development'.

Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.


  Contract: DataComplianceContract
    ✓ Register consumer (69ms)
    ✓ Register provider (49ms)
    ✓ Transfer Data From Provider A to Provider B (52ms)
    ✓ Transfer Data From Consumer to Provider A (50ms)


  4 passing (319ms)
```

Figure 5 : Testing the contract and network

The DataComplianceContract is the smart contract deployed on the network and automatically validates a condition and determines if data is allowed to be transferred from Consumer to Provider. All the members are initially registered on the network which is shown in Figure 6.

```
Getting deployed version of DataComplianceContract...
Registering Consumer on the network
Transaction: 0x16ce4eae0a13c6059f051172c889ea3c4e2f63cdf528c0710e2150d696a11d50
Finished!
Using network 'development'.

Getting deployed version of DataComplianceContract...
Registering Provider on the network
Transaction: 0xc5babf99dde004c2c463e904443223d86d2695037b2a779bc2419b7f4d67a8dd
Finished!
```

Figure 6: Register members on the network

```
Tranfer Data from Microsoft to LinkedIn
Using network 'development'.

Getting deployed version of DataComplianceContract...
Transfering user data from Microsoft to LinkedIn
Transaction: 0xbf57ad652b39bcb4706a7221b5fc67531e387c81c6f78572aa311d0d3cb06
ddb
Transaction Successful!
```

Figure 7(a): Transferring Data from Microsoft to LinkedIn

```
 Transaction: 0xbf57ad652b39bcb4706a7221b5fc67531e387c81c6f78572aa311d0d
3cb06ddb
  Gas usage: 132037
  Block Number: 9
  Block Time: Tue Mar 24 2020 11:55:53 GMT-0400 (Eastern Daylight Time)
```

Figure 7(b): Transaction Receipt for Successful Data Transfer

For every transaction on the network a SPARQL query is executed on the privacy ontology to check for permission. Based on the checks defined in the smart contract, the transaction is executed. In the case of Figure 7(a) and Figure 7(b), Microsoft is transferring PII data to LinkedIn. As the regulatory compliance allows Microsoft to transfer data to LinkedIn in the awareness of the consumer , the transaction is successful ensuring Article 24 i.e. Processing is performed in accordance with regulation.

```
Using network 'development'.

Getting deployed version of DataComplianceContract...
Transfering user data from Microsoft to Google
Transaction Failed
Provider is not allowed to transfer Data
```

Figure 8(a): Transferring Data from Microsoft to Google

```
Gas usage: 25404
Block Number: 12
Block Time: Tue Mar 24 2020 12:01:36 GMT-0400 (Eastern Daylight Time)
Runtime Error: revert
Revert reason: Provider is not allowed to transfer data
```

Figure 8(b): Transaction Receipt for Failed Data Transfer

When it comes to Microsoft transferring PII data to Google, the transaction is denied in Fig 8(a) and 8(b). The regulatory compliance of Microsoft does not allow the transfer of data to any other company other than its own companies. When the transaction is initiated, the SPARQL query checks with the policy ontology for permission, ensuring privacy based on the GDPR article 24 which says it is the responsibility of the controller to process data in accordance with the regulation. All the transactions are recorded on the blockchain network which ensures article 30.

V. CONCLUSION

Many regulatory bodies are releasing new data protection laws every year to ensure data security. These regulations are in text format and it needs a high amount of human time and effort to get an assurance of compliance over the network. We believe that the semantically rich, machine-readable knowledge graph manages to capture most of the regulations which assist in automating an organization's data compliance process. The personal data of a consumer is private and should not be shared without the consent of the consumer. Our architecture uses the blockchain platform to help maintain this scenario where the consumer will always have access to their data and track the lifecycle of the data. As the blockchain platform is decentralized, making regulatory decisions about collecting, storing and sharing of personal data is much easier.

Our proposed system addresses the problems of data tracking or PII tracking plus the data stored in an encrypted format ensures security. This system was only implemented using Ethereum as blockchain technology, but we plan on exploring other blockchain frameworks like Hyperledger Iroha, Hyperledger Indy and tools like Hyperledger Composer to create a larger network and better processing capabilities. Huge development is in progress in the blockchain industry to integrate the GDPR, similarly, we plan to cover a few GDPR issues like ‘Privacy by Design’, ‘Processing’, ‘Right to Access’, ‘Third Countries’[29] in our future work.

VI. ACKNOWLEDGEMENTS

This research was partially supported by a DoD supplement to the NSF award# 1747724, Phase I IUCRC UMBC: Center for Accelerated Real time Analytics (CARTA)

VII. REFERENCES

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Consulted, 1(2012):28, 2008.

[2] EUROPEAN COMMISSION. Commission proposes a comprehensive reform of data protection rules to increase users' control of their data and to cut costs for businesses. 2012.

[3] Karuna Pande Joshi et al., "Automating Cloud Services Lifecycle through Semantic technologies", Article, IEEE Transactions on Service Computing, January 2014,K. Elissa, "Title of paper if known," unpublished.

[4] IT Pro Portal. (2018). PCI and GDPR: How to be cross-compliant. [online] Available at: https://www.itproportal.com/features/pci-and-gdpr-how-to-be-cross-compliant/ [Accessed 17 Aug. 2018].

[5] OWL: Web Ontology Language, Sean Bechhofer, https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_1073

[6] S. Soderland, Learning to extract text-based information from the world wide web, in KDD, vol. 97, 1997, pp. 251254

[7] Understanding the 12 Requirements of PCI DSS -Practical steps to achieve and maintain compliance, OpinionPiece[Online].www.dimensiondata.com/globalpresence. [Accessed: 21- Feb- 2016]

[8] K. P. Joshi and C. Pearce, "Automating Cloud Service Level Agreements Using Semantic Technologies," 2015 IEEE International Conference on Cloud Engineering, Tempe, AZ,2015, pp. 416-421, doi: 10.1109/IC2E.2015.63

[9] EU GDPR Portal. (2018). GDPR Glossary of Terms. [online] Available at: https://www.eugdpr.org/glossary-of-terms.html [Accessed 17 Aug. 2018].

[10] "General Data Protection Regulation (GDPR) – Final text neatly arranged." General Data Protection Regulation (GDPR), gdpr-info.eu/

[11] Nasr Al-Zaben , Md Mehedi Hassan Onik , Jinhong Yang , Nam-Yong Lee , Chul-Soo Kim. General Data Protection Regulation Complied Blockchain Architecture for Personally Identifiable Information Management

[12] OAuth Protocol, https://tools.ietf.org/html/rfc6749

[13] The truth about Blockchain, Harvard Business Reviews, https://hbr.org/2017/01/the-truth-about- blockchain.

[14] http://bitfury.com/content/5-white-papers-research/bitfury_white_paper_on_blockchain_auditability.pdf

[15] Karuna P. Joshi et al., Semantic Approach to Automating Management of Big Data Privacy Policies, in proceedings of IEEE Big Data, 2016.

[16] Guy Zyskind, Oz Nathan, Alex 'Sandy' Pentland. Decentralizing Privacy: Using Blockchain to Protect Personal Data.2015 IEEE CS Security and Privacy Workshops.

[17] Wang,Huaiqing,Chen,Kun,Xu, Dongming. A maturity model for blockchain adoption, Financial Innovation, Dec 2016.

[18] D. McGuinness, F. Van Harmelen, et al., OWL web ontology language overview, W3C recommendation, World Wide Web Consortium, 2004.

[19] Chinmay Saraf, Siddharth Sabadra. Blockchain Compendium. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8376323

[20] Bin Yu1 et. Al . Platform-independent Secure Blockchain-Based Voting System. https://eprint.iacr.org/2018/657.pdf

[21] JIALU HAO1 et al . Efficient Attribute-based Access Control with Authorized Search in Cloud Storage. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8672564

[22] Stephen Kirkman. A Data Movement Policy Framework for Improving Trust in the Cloud Using Smart Contracts and Blockchains. https://ieeexplore.ieee.org/document/8360339

[23] Shinsaku Kiyomoto, Mohammad Shahriar Rahman, Anirban Basu. On Blockchain-Based Anonymized Dataset DistributionPlatform. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7965711

[24] Shangping Wang, Yinglong Zhang , Yaling Zhang. A Blockchain-Based Framework for Data Sharing With Fine-Grained Access Control in Decentralized Storage Systems. https://ieeexplore.ieee.org/document/8400511

[25] Kristin B. Cornelius, Department of Information Studies, University of California, Los Angeles, United States of America. Standard form and contracts and a smart contract future. https://policyreview.info/articles/analysis/standard-form-contracts-and-smart-contract-future

[26] Chun-Feng Liao. Toward A Service Platform for Developing Smart Contracts on Blockchain in BDD and TDD styles. https://ieeexplore.ieee.org/document/8241535

[27] https://hyperledger-fabric.readthedocs.io/en/release-1.4/smartcontract/smartcontract.html

[28] Emanuel Regnath, Sebastian Steinhorst. SmaCoNat: Smart Contracts in Natural Language.https://s-steinhorst.github.io/PDF/2018-FDL-SmaCoNat%20-%20Smart%20Contracts%20in%20Natural%20Language.pdf

[29] https://gdpr-info.eu/

[30] https://bitsonblocks.net/2016/02/29/a-gentle-introduction-to-immutability-of-blockchains/

[31] Lavanya Elluri, Ankur Nagar, Karuna Pande Joshi. An Integrated Knowledge Graph to Automate GDPR and PCI DSS Compliance.

[32] Lavanya Elluri and Karuna Pande Joshi. A Knowledge Representation of Cloud Data controls for EU GDPR Compliance.

[33] Agniva Banerjee and Dr. Karuna Pande Joshi. Link Before You Share: Managing Privacy Policies through Blockchain.

[34] https://www.trufflesuite.com/docs/truffle/testing/writing-tests-in-javascript

[35] Srishty Saha, Karuna P. Joshi, Renee Frank, Michael Aebig, Jiayong Lin. Automated Knowledge Extraction from the Federal Acquisition Regulations System https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8258353

[36] TheHIPAAPrivacy Rule, https://www.hhs.gov/hipaa/for-professionals/privacy/index.html

[37] Payment Card Industry (PCI) Data Security Standard,Version3.2,https://www.pcisecuritystandards.org/document_library, August 2019